Multiple Linear Regression

A regression with two or more explanatory variables is called a multiple regression. Rather than modeling the mean response as a straight line, as in simple regression, it is now modeled as a function of several explanatory variables. The function lm can be used to perform multiple linear regression in R and much of the syntax is the same as that used for fitting simple linear regression models. To perform multiple linear regression with *p* explanatory variables use the command:

lm(response ~ explanatory_1 + explanatory_2 + … + explanatory_p)

Here the terms *response* and *explanatory_i* in the function should be replaced by the names of the response and explanatory variables, respectively, used in the analysis.

Ex. Data was collected on 100 houses recently sold in a city. It consisted of the sales price (in $), house size (in square feet), the number of bedrooms, the number of bathrooms, the lot size (in square feet) and the annual real estate tax (in $).

The following program reads in the data.

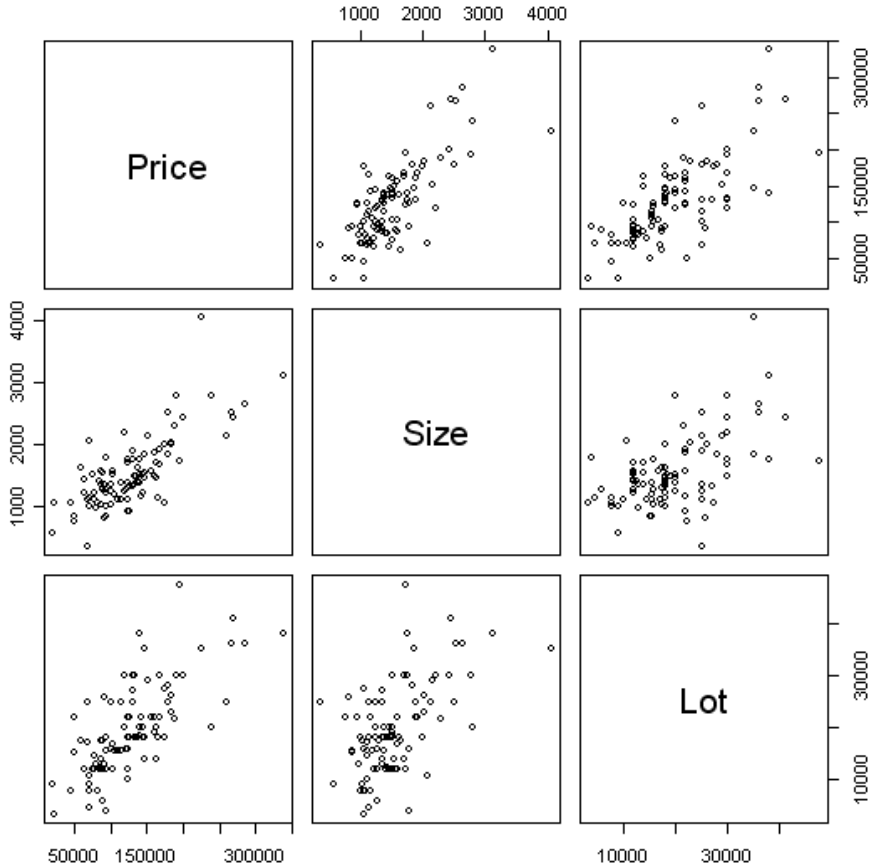
```
> Housing = read.table("C:/Users/Martin/Documents/W2024/housing.txt",
header=TRUE)
> Housing
    Taxes Bedrooms Baths  Price Size   Lot
1    1360        3   2.0 145000 1240 18000
2    1050        1   1.0  68000  370 25000
…..
99    1770        3   2.0  88400 1560 12000
100  1430        3   2.0 127200 1340 18000
```

Suppose we are only interested in working with a subset of the variables (e.g., “Price”, “Size” and “Lot”). It is possible (but not necessary) to construct a new data frame consisting solely of these values using the commands:

```
> myvars = c("Price", "Size", "Lot")
> Housing2 = Housing[myvars]
> Housing2
     Price Size   Lot
1   145000 1240 18000
2    68000  370 25000
……..
99   88400 1560 12000
100 127200 1340 18000
```

Before fitting our regression model we want to investigate how the variables are related to one another. We can do this graphically by constructing scatter plots of all pair-wise combinations of variables in the data frame. This can be done by typing:

```
> plot(Housing2)
```



To fit a multiple linear regression model with price as the response variable and size and lot as the explanatory variables, use the command:

```
> results = lm(Price ~ Size + Lot, data=Housing)
> results

Call:
lm(formula = Price ~ Size + Lot, data = Housing)
Coefficients:
(Intercept)        Size         Lot
 -10535.951      53.779       2.840
```

This output indicates that the fitted value is given by $\hat{y} = -10536 + 53.8x_1 + 2.8x_2$

Inference in the multiple regression setting is typically performed in a number of steps. We begin by testing whether the explanatory variables collectively have an effect on the response variable, i.e.

$$H_0 : \beta_1 = \beta_2 = \ldots \beta_p = 0$$

If we can reject this hypothesis, we continue by testing whether the individual regression coefficients are significant while controlling for the other variables in the model.

We can access the results of each test by typing:

```
> summary(results)

Call:
lm(formula = Price ~ Size + Lot, data = Housing)
Residuals:
   Min      1Q  Median   3Q     Max
-81681 -19926   2530  17972  84978
Coefficients:
             Estimate      Std. Error    t value   Pr(>|t|)
(Intercept) -1.054e+04   9.436e+03   -1.117     0.267
Size          5.378e+01   6.529e+00    8.237     8.39e-13 ***
Lot           2.840e+00   4.267e-01     6.656    1.68e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 30590 on 97 degrees of freedom
Multiple R-squared: 0.7114,     Adjusted R-squared: 0.7054
F-statistic: 119.5 on 2 and 97 DF,  p-value: < 2.2e-16
```

The output shows that F = 119.5 (p < 2.2e-16), indicating that we should clearly reject the null hypothesis that the variables Size and Lot collectively have no effect on Price. The results also show that the variable Size is significant controlling for the variable Lot (p = 8.39e-13), as is Lot controlling for the variable Size (p=1.68e-09). In addition, the output also shows that R^2 = 0.7114 and $R^2_{adjusted}$ = 0.7054.

B. Testing a subset of variables using a partial F-test

Sometimes we are interested in simultaneously testing whether a certain subset of the coefficients are equal to 0 (e.g. $\beta_3 = \beta_4 = 0$). We can do this using a partial F-test. This test involves comparing the SSE from a reduced model (excluding the parameters we hypothesis are equal to zero) with the SSE from the full model (including all of the parameters).

In R we can perform partial F-tests by fitting both the reduced and full models separately and thereafter comparing them using the anova function.

Ex. Suppose we include the variables bedroom, bath, size and lot in our model and are interested in testing whether the number of bedrooms and bathrooms are significant after taking size and lot into consideration.

The following code performs the partial F-test:

```
> reduced = lm(Price ~ Size + Lot, data=Housing)       # Reduced model
> full = lm(Price ~ Size + Lot + Bedrooms + Baths, data=Housing)   # Full Model

> anova(reduced, full)                                 # Compare the models

Analysis of Variance Table
Model 1: Price ~ Size + Lot
Model 2: Price ~ Size + Lot + Bedrooms + Baths
  Res.Df    RSS          Df   Sum of Sq      F          Pr(>F)
1     97      9.0756e+10
2     95      8.5672e+10   2   5083798629    2.8186   0.06469 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output shows the results of the partial F-test. Since F=2.82 (p-value=0.0647) we cannot reject the null hypothesis ($\beta_3 = \beta_4 = 0$) at the 5% level of significance. It appears that the variables Bedrooms and Baths do not contribute significant information to the sales price once the variables Size and Lot have been taken into consideration.

C. Confidence and Prediction Intervals

We often use our regression models to estimate the mean response or predict future values of the response variable for certain values of the response variables. The function predict() can be used to make both confidence intervals for the mean response and prediction intervals. To make confidence intervals for the mean response use the option interval="confidence". To make a prediction interval use the option interval="prediction". By default this makes 95% confidence and prediction intervals. If you instead want to make a 99% confidence or prediction interval use the option level=0.99.

Ex. Obtain a 95% confidence interval for the mean sales price of houses whose size is 1,000 square feet and lot size is 20,000 square feet.

```
> results = lm(Price ~ Size + Lot, data=Housing)
```

```
> predict(results,data.frame(Size=1000, Lot=20000),interval="confidence")
        fit       lwr      upr
[1,] 100050.1  90711.45  109388.7
```

A 95% confidence interval is given by (90711, 109389)

Ex. Obtain a 95% prediction interval for the sales price of a particular house whose size is 1,000 square feet and lot size is 20,000 square feet.

```
> predict(results,data.frame(Size=1000, Lot=20000),interval="prediction")
        fit       lwr      upr
[1,] 100050.1  38627.08  161473.0
```

A 95% prediction interval is given by (38627, 161473). Note that this is quite a bit wider than the confidence interval, indicating that the variation about the mean is fairly large.