Linear Regression in R

Hao Zhang

A linear regression model takes the following general form

\[ y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon \]

where \( y \) is the response or dependent variable, \( x_1, \ldots, x_p \) are the explanatory or independent variables, \( \epsilon \) is a random term with mean 0. Given \( n \) observations of each of the variables \( y, x_1, \ldots, x_p \), the model becomes

\[ y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i, \ i = 1, \ldots, n \]

where the error terms \( \epsilon_i \) are assumed to be i.i.d. \( N(0, \sigma^2) \).

The statistical analysis for such a linear model usually includes the following

- Estimate the parameters \( \beta_1, \ldots, \beta_p \) and \( \sigma^2 \).
- Test the hypothesis that \( \beta_i = 0 \) for a particular \( i \).
- Test the hypothesis that a subset of the coefficients \( \beta_i \) equals 0, e.g., \( \beta_1 = \beta_2 = 0 \).
- Check the model assumptions
- Prediction of response variable for some given values of the explanatory variables.

The following are some R codes for the analysis. We will use the built-in dataset state.x77 to demonstrate.

```R
> state=as.data.frame(state.x77)
> str(state) # A compact display of the structure of the data.
'data.frame': 50 obs. of 8 variables:
$ Population: num 3615 365 2212 2110 21198 ... 
$ Income : num 3624 6315 4530 3378 5114 ... 
$ Illiteracy: num 2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ... 
$ Life Exp : num 69 69.3 70.5 70.7 71.7 ... 
$ Murder : num 15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ... 
$ HS Grad : num 41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ... 
$ Frost : num 20 152 15 65 20 166 139 103 11 60 ... 
$ Area : num 50708 566432 113417 51945 156361 ... 
> colnames(state)
> colnames(state)[4]="LifeExp"
> colnames(state)[6]="HSGrad"
>
> # Alternatively, you can remove the spaces in all names by
> colnames(state)=sapply(colnames(state), function(x)(gsub(" ", "", x)))
> > pairs(state)
> > state.lm1=lm(formula = LifeExp ~ Population + Income + Illiteracy + Murder +
+    HSGrad + Frost + Area, data = state)
> # or
> state.lm1= lm(LifeExp ~., data=state)
> summary(state.lm1)

Call:
 lm(formula = LifeExp ~ ., data = state)

Residuals:    
 Min 1Q Median 3Q Max
-1.48895 -0.51232 -0.02747 0.57002 1.49447

Coefficients:    
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.094e+01 1.748e+00 40.586 < 2e-16 ***
Population 5.180e-05 2.919e-05 1.775 0.0832 .
Income -2.180e-05 2.444e-04 -0.089 0.9293
Illiteracy 3.382e-02 3.663e-01 0.092 0.9269
Murder -3.011e-01 4.662e-02 -6.459 8.68e-08 ***
HSGrad 4.893e-02 2.332e-02 2.098 0.0420 *
Frost -5.735e-03 3.143e-03 -1.825 0.0752 .
Area -7.383e-08 1.668e-06 -0.044 0.9649
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.7448 on 42 degrees of freedom
Multiple R-squared: 0.7362,    Adjusted R-squared: 0.6922
F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10

> state.lm2=update(state.lm1, .~.-Income-Illiteracy-Area)
> summary(state.lm2)

Call:
 lm(formula = LifeExp ~ Population + Murder + HSGrad + Frost,  
 data = state)

Residuals:    
 Min 1Q Median 3Q Max
-1.47095 -0.53464 -0.03701 0.57621 1.50683
Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | 7.103e+01 | 9.529e-01 | 74.542 | < 2e-16 *** |
| Population | 5.014e-05 | 2.512e-05 | 1.996 | 0.05201 . |
| Murder | -3.001e-01 | 3.661e-02 | -8.199 | 1.77e-10 *** |
| HSGrad | 4.658e-02 | 1.483e-02 | 3.142 | 0.00297 ** |
| Frost | -5.943e-03 | 2.421e-03 | -2.455 | 0.01802 * |

---

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.7197 on 45 degrees of freedom
Multiple R-squared: 0.736, Adjusted R-squared: 0.7126
F-statistic: 31.37 on 4 and 45 DF, p-value: 1.696e-12

> anova(state.lm2, state.lm1)
Analysis of Variance Table

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>RSS</td>
</tr>
<tr>
<td>1</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
</tr>
</tbody>
</table>

> plot(state.lm2$fitted, state.lm2$residuals, xlab="Fitted value", ylab="Residual")
> qqnorm(state.lm2$residuals)
> abline(0, 1)
> hist(state.lm2$residuals, prob=T)
> confint(state.lm2) #This provides confidence intervals for the coefficients.

| (Intercept) | 6.910798e+01 | 72.9462729104 |
| Population | -4.543308e-07 | 0.0001007343 |
| Murder | -3.738840e-01 | -0.2264135705 |
| HSGrad | 1.671901e-02 | 0.0764454870 |
| Frost | -5.943e-03 | 2.421e-03 |

We see that the residuals show some departure from a normal distribution.

We will look at another example. This dataset is provided in the MASS library. The data contains the time (in minutes), distance in miles traveled and the total height gained during the route (in feet) of 35 Scottish hill races. We will identify the influential cases/observations. A case is influential if deletion of the case from the data leads to a large change in the estimates of the model. Cook's distance is a measure of influence. Those will large Cook's distances (usually greater than 1) are considered to be influential or outliers.
> library(MASS)
> pairs(hills) # Graph not shown here.
> hills.lm = lm(time ~ dist + climb, data = hills)
> summary(hills.lm)

Call:
  lm(formula = time ~ dist + climb, data = hills)

Residuals:
     Min      1Q  Median      3Q     Max
-16.215  -7.129  -1.186  2.371  65.121

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.992039  4.302734  -2.090  0.0447 *
dist          6.217956  0.601148   10.343 9.86e-12 ***
climb         0.011048  0.002051   5.387  6.45e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 14.68 on 32 degrees of freedom
Multiple R-squared: 0.9191,  Adjusted R-squared: 0.914
F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16

> par(mfrow = c(2, 2))
> plot(hills.lm)
> cooks.distance(hills.lm)

Greenmantle Carnethy Craig Dunain Ben Rha
5.127185e-04 4.875401e-03 1.365422e-03 6.433010e-05
Ben Lomond Goatfell Bens of Jura Cairnpapple
1.474139e-02 5.129264e-05 1.893349e+00 4.277564e-05
Scolty Traprain Lairig Ghru Dollar
9.519595e-05 1.405256e-03 2.105214e-01 9.612212e-05
Lomonds Cairn Table Eildon Two Cairngorm
4.703839e-03 9.339448e-03 4.834282e-03 1.490749e-02
Seven Hills Knock Hill Black Hill Creag Beag
2.385559e-03 4.071560e-01 1.026539e-02 4.177135e-05
Kildcon Hill Meall Ant-Suidhe Half Ben Nevis Cow Hill
8.700051e-04 1.716152e-04 1.661162e-03 2.107872e-03
N Berwick Law Creag Dubh Burnswark Largo Law
4.986386e-04 1.317565e-02 5.113116e-04 1.017978e-03
Criefel Acmony Ben Nevis Knockfarrel
3.289579e-06 3.669350e-03 6.412250e-02 5.311898e-04
Two Breweries Cockleroi Moffat Chase
3.769491e-02 5.293489e-06 5.242223e-02

> sort(cooks.distance(hills.lm), decreasing = T)
Note the changes in the estimates!

Now you can use the model to make prediction for the response variable (time) for given values of the explanatory variables.
Lastly, we will apply bootstrap to a linear regression model, which is used when the normality assumption is not valid. Normal probability plot reveals a clear departure from normality.

```r
hist(resid(state.lm2))
qqnorm(resid(state.lm2))
abline(0, 1) # clear departure from normal distribution
```

To Bootstrap a linear model is essentially bootstrapping the residuals. For each bootstrap sample of the residuals, construct the new responses by adding the fitted value. Then fit the linear model to the new dataset.

```r
result=matrix(nrow=500, ncol=length(coef(state.lm2)))
for(i in 1:500){
d=state
d$LifeExp=fitted(state.lm2)+sample(resid(state.lm2), replace=T)
result[i,]=coef(update(state.lm2, data=d))
}
k=apply(result, 2, quantile, p=c(0.025, 0.975))
#Percentile CI
k
# basic CI
cbind(2*colMeans(result)-k[2,],2*colMeans(result)-k[1,])
```

### Influential observations and Cook’s distance

A data point is influential to the fitted response if omitting it would result in a large change in fitted value. Cook’s distance is commonly used to detect the influential data points.

Cook’s distance measures the effect of deleting a given observation. Data points with large residuals and/or high leverage may distort the outcome and accuracy of a regression. Points with a large Cook’s distance are considered to merit closer examination in the analysis. It is calculated as:

$$ D_i = \frac{\sum_{j=1}^n (\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \text{ MSE}}, $$

where: $\hat{Y}_j$ is the prediction from the full regression model for observation $j$; $\hat{Y}_{j(i)}$ is the prediction for observation $j$ from a refitted regression model in which observation $i$ has been omitted; $p$ is the number of fitted parameters in the model; MSE is the mean square error of the regression model.

It can be shown

$$ D_i = \frac{e_i^2}{p \text{ MSE}} \left[ \frac{h_{ii}}{(1 - h_{ii})^2} \right] $$

where $h_{ii}$ is the leverage, i.e., the i-th diagonal element of the hat matrix $X (X^T X)^{-1} X^T$; $e_i$ is the residual (i.e., the difference between the observed value and the value fitted by the proposed model).

There are different opinions regarding what cut-off values to use for spotting highly influential points. A simple guideline of $D_i > 1$ has been suggested. Others have indicated that $D_i > 4/n$, where $n$ is the number of observations, might be used.