Additional Topics in Simple Linear Regression

In this tutorial we discuss additional topics related to simple linear regression. We begin by considering regression from the perspective of analysis of variance (ANOVA). This way of looking at the data will become increasingly important when we start discussing multiple regression models. Next, we discuss fitting a regression line that is constrained to go through the origin.

A. Regression and ANOVA

Analysis of Variance (ANOVA) summarizes information about different sources of variation in the data. In ANOVA we study the variation between individual values of the response variable and their mean. In regression this variation depends on (a) the distance of the regression line to the mean, i.e. the variation explained by the model; and (b) the distance of the observation to the regression line, i.e. the variation not explained by the model. This decomposition can be obtained by using the command ANOVA(results), where results is the output of a previous regression analysis.

We have already looked at hypothesis tests for H_0: $\beta_1=0$ vs. H_a: $\beta_1\neq0$ using a t-test. An equivalent test can be performed in the ANOVA setting using the F-statistic F=MSR/MSE.

<u>Ex.</u> A pediatrician wants to study the relationship between a child's height and their head circumference (both measured in inches). She selects a SRS of 11 three-year old children and obtains the following data.

Suppose the data is contained in a text file called "Kids.txt" that does not have a header (i.e. information about the variable names in the first row of the text file). We will therefore want to give names to the variables when reading in the data set into R. In this particular case, we can do this by including the option col.names=c("Height", "Circ") when using the function read.table(). This will give the variable contained in the first column the name Height, and the variable in the second column the name Circ. If we had not included this option R would have given them the default names v1 and v2, respectively.

```
> Dat = read.table("Kids.txt",header=FALSE,col.names=c("Height", "Circ"))
> Dat
  Height Circ
1   27.75 17.5
2   24.50 17.1
…..
10  26.75 17.5
11  27.50 17.5

> attach(Dat)
```

After reading in the data set we fit a simple linear regression model and obtain the ANOVA decomposition.

```
> results = lm(Circ ~ Dat)
> anova(results)
Analysis of Variance Table
Response: Dat$Circ
                 Df    Sum Sq    Mean Sq    F value   Pr(>F)
Dat$Height        1    0.39993   0.39993    43.958    9.59e-05 ***
Residuals         9    0.08188   0.00910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the ANOVA table, we see that the *F*-statistic is equal to 43.958 with a p-value of 9.59e-05. Hence, there is strong evidence that β_1 is not equal to zero.

Compare this to the results obtained in the previous tutorial.

```
> summary(results)
Call:
lm(formula = Dat$Circ ~ Dat$Height)
Residuals:
     Min       1Q   Median       3Q      Max
-0.16148 -0.05842 -0.01831  0.06442  0.12989
Coefficients:
                 Estimate    Std. Error   t value   Pr(>|t|)
(Intercept)      12.49317    0.72968      17.12     3.56e-08 ***
Dat$Height       0.18273     0.02756      6.63      9.59e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.09538 on 9 degrees of freedom
Multiple R-Squared: 0.8301,     Adjusted R-squared: 0.8112
F-statistic: 43.96 on 1 and 9 DF,  p-value: 9.59e-05
```

From the output we see that we can clearly reject the null hypothesis of no linear relationship between height and head circumference (p=9.59e-05). Note this is the same value obtained using the ANOVA F-test.

The square of the correlation, r^2=SSR/SST is equal to the ratio of the regression sum of squares to the total sum of squares. It measures the fraction of variability in the data explained by the regression model. Here the r^2 term is equal to 0.83, indicating that 83% of the variability in the response is explained by the explanatory variable.

B. Regression through the origin

In some instances we may want to constrain our regression line to go through the origin. One example would be if we were measuring the distance traveled as a function of time, as at time 0 we would expect to have traveled a distance 0. In this case we can use a no-intercept regression model: $Y_i = \beta_1 X_i + \varepsilon_i$ where the ε_i are independent $N(0,\sigma)$. This model forces the regression line to go through the point (0,0).

We can perform regression through the origin using either of the following commands: lm(response ~ 0 + explanatory) or lm(response ~ explanatory -1) where response indicates the response variable and explanatory the explanatory variable.

Ex. A plumbing supply company was interested in studying the relationship between the number of jobs performed (X) and total variable labor cost (Y, in thousands of dollars) at each of its 12 warehouses. Fit the data using regression through the origin.

We begin by reading the data and fitting a regression model.

```
> Units = c(20, 196, 115, 50, 122, 100, 33, 154, 80, 147, 182, 160)
> Cost = c(114, 921, 560, 245, 575, 475, 138, 727, 375, 670, 828, 762)
> Dat = data.frame(Units,Cost)
> Results = lm(Cost ~ Units -1)
> Results
Call:
lm(formula = Cost ~ Units - 1)
Coefficients:
Units
4.685
```

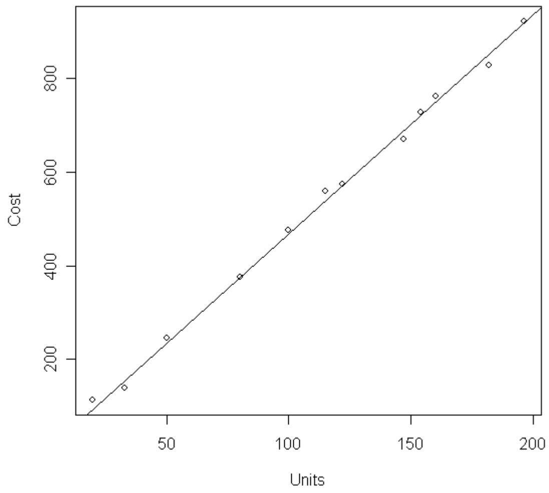
According to the output the estimated regression function is yhat=4.685X. This implies that for each additional job performed the labor costs increases by 4.685 thousand dollars, or $4685.

If we want to make a 90% confidence interval for the slope we can use the function confint() together with the option level=0.90.

```
> confint(Results,level=0.90)
            5 %       95 %
Units 4.623846 4.746702
```

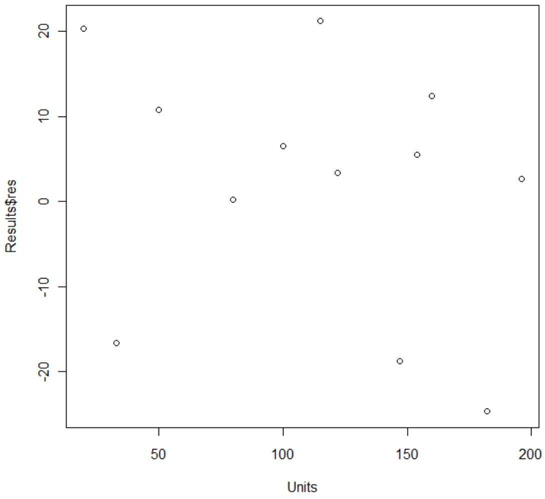
We can visualize the fitted regression function overlayed on a scatter plot of the two variables using the following commands:

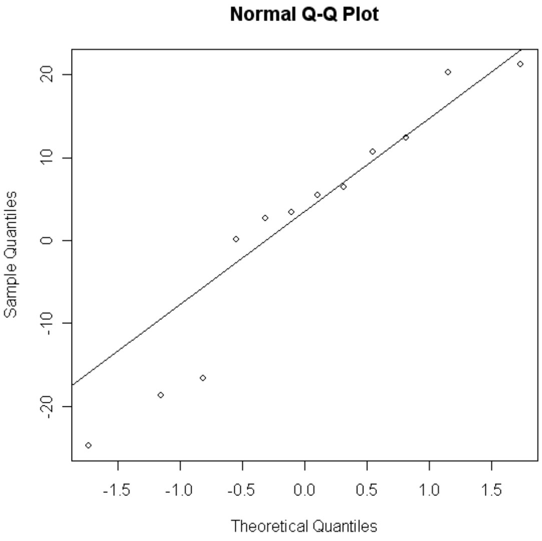
```
> plot(Units,Cost)
> abline(Results)
```



To further access the quality of the fit we make residual plots and normal probability plots.

```
> plot(Units,Results$res)
> qqnorm(Results$res)
> qqline(Results$res)
```





The residuals appear to be randomly scattered around the mean with constant variance. The normal probability plot indicates the three smallest residuals are somewhat smaller than expected, but probably not to such a degree that they will adversely affect the model fit.

To test significance of slope we use the summary() command:

```
> summary(Results)
Call:
lm(formula = Cost ~ Units - 1)
Residuals:
    Min       1Q    Median    3Q       Max
-24.720  -4.020   4.432  11.141  21.193
Coefficients:
        Estimate   Std. Error  t value   Pr(>|t|)
Units  4.68527    0.03421    137.0    <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 14.95 on 11 degrees of freedom
Multiple R-Squared: 0.9994,     Adjusted R-squared: 0.9994
F-statistic: 1.876e+04 on 1 and 11 DF,  p-value: < 2.2e-16
```

Judging from the output there is extremely strong evidence of a linear relationship between the variables Cost and Units (p<2e-16).

Suppose we are interested in obtaining a 90% confidence intervals for the mean cost when there are 100 jobs. The function predict() can be used to make confidence intervals for the mean response using the options interval=”confidence” and level=0.90.

```
> predict(Results, data.frame(Units =100), interval="confidence", level=0.90)
         fit                 lwr            upr
[1,] 468.5274    462.3846    474.6702
```

The 90% confidence interval is given by (462.39, 474.67).

Finally, to obtain a 95% prediction intervals for the predicted cost when there are 100 jobs. The function predict() can be used to make prediction interval using the options interval=”prediction” and level=0.90.

```
> predict(Results,data.frame(Units =100),interval="prediction",level=0.90)
         fit                 lwr           upr
[1,] 468.5274    440.9898    496.065
```

The 90% prediction interval is given by (440.99, 496.06).