Assignment Lecture10

Zepeng Mu

3/29/2017

Introduction

Analysis of Covariance (ANCOVA) is typically used when the value of a continuous variable y dependent on at least one continuous variable (known as independent variance) and at least one categorical variable (known as covriance). The basic concept is to perform linear regression in each group independently.

Assumption for ANCOVA

1. Normality of sampling distribution
2. Homogeneity of variance
3. Linearity
4. Homogeneity of regression
5. Each CV is measured without error
6. No outliers
7. No multicollinearity/Singularity

Data set

Here I used several generated data to perform ANCOVA. To begin with, X values is generated. Since I did not want the X values to be evenly distributed, I used the `sample()` function and picked the unique value.

```
a = 2
xValue1 = unique(sample(seq(5, 20, by = 0.2), 60, rep = TRUE)); xValue1
```

```
##  [1] 10.0 15.6  6.4  7.8 16.2  9.0  8.4  9.8 18.6 19.8  5.4 17.8  7.4  5.6
## [15] 14.2 19.6 11.2 16.6  5.8 18.0 12.8 17.4 13.4 16.8 10.6  7.0  7.2 15.2
## [29] 20.0  8.6 16.0 13.8 10.2 10.4  9.2 11.4  6.2  8.2 12.2 19.2  5.0 18.2
## [43]  6.8
```

Here, I could assume that the value of X has no error. The next step is to get Y values. I first got exact Y values using a linear function:

```
b1 = 17
y1 = a * xValue1 + b1; y1
```

```
##  [1] 37.0 48.2 29.8 32.6 49.4 35.0 33.8 36.6 54.2 56.6 27.8 52.6 31.8 28.2
## [15] 45.4 56.2 39.4 50.2 28.6 53.0 42.6 51.8 43.8 50.6 38.2 31.0 31.4 47.4
## [29] 57.0 34.2 49.0 44.6 37.4 37.8 35.4 39.8 29.4 33.4 41.4 55.4 27.0 53.4
## [43] 30.6
```

Note that by doing so I actually satisfied the third assumption mentioned above, that is, the actual relationship is linear. Then, I need to make some 'errors' and to get the values of Y that are not perfect linear to X:

```
yErr1 = sample(seq(-6.5, 5, by = 0.2), length(y1), rep = TRUE); yErr1
```

```
##  [1]  0.5  3.3 -0.9 -6.5  2.7  4.7  4.7  2.1 -0.3 -4.5 -2.9  3.5 -4.1 -2.5
## [15] -0.1 -0.1 -3.1  0.9  4.9  0.3 -1.5 -2.7 -5.9  4.1 -5.7  4.1  1.3  4.9
## [29] -1.9 -6.5 -3.9  1.5  1.3 -0.3 -4.3  1.5  3.9 -0.9 -2.7  3.9  1.9 -6.3
## [43] -0.5
```

```
yValue1 = y1 + yErr1; yValue1
```

```
##  [1] 37.5 51.5 28.9 26.1 52.1 39.7 38.5 38.7 53.9 52.1 24.9 56.1 27.7 25.7
## [15] 45.3 56.1 36.3 51.1 33.5 53.3 41.1 49.1 37.9 54.7 32.5 35.1 32.7 52.3
## [29] 55.1 27.7 45.1 46.1 38.7 37.5 31.1 41.3 33.3 32.5 38.7 59.3 28.9 47.1
## [43] 30.1
```

Then, another set of data with identical slope but different intercepts are generated. Thus, the fourth and the last assumption mentioned above are satisfied.

```
xValue2 = unique(sample(seq(5, 20, by = 0.2), 60, rep = TRUE)); xValue2
```

```
##  [1]  9.4  7.8 15.8  8.0 11.4 11.8  5.8  8.8 10.0  9.2  6.6 10.2 10.6 18.4
## [15] 19.4 17.4 17.2 18.8 16.8 14.8  9.6 19.2 17.0 16.0  5.6 18.6 17.8 11.2
## [29] 16.2 14.2  8.4  5.2  5.4  7.2 10.4  9.0 13.6 13.2
```

```
b2 = 2
y2 = a * xValue2 + b2; y2
```

```
##  [1] 20.8 17.6 33.6 18.0 24.8 25.6 13.6 19.6 22.0 20.4 15.2 22.4 23.2 38.8
## [15] 40.8 36.8 36.4 39.6 35.6 31.6 21.2 40.4 36.0 34.0 13.2 39.2 37.6 24.4
## [29] 34.4 30.4 18.8 12.4 12.8 16.4 22.8 20.0 29.2 28.4
```

```
yErr2 = sample(seq(-5, 6.5, by = 0.2), length(y2), rep = TRUE); yErr2
```

```
##  [1]  4.4  2.2  0.6  4.6  2.6 -0.6  2.2  1.4  2.6  4.6  3.8  5.2  1.0  2.4
## [15]  2.6  1.8  5.6  3.0 -1.2 -0.6 -3.0 -1.2  6.0 -0.6 -4.8  4.2  0.6  2.6
## [29] -4.2  4.4 -1.0  0.4  1.2 -0.4  4.8  2.4 -0.4  0.2
```

```
yValue2 = y2 + yErr2; yValue2
```

```
##  [1] 25.2 19.8 34.2 22.6 27.4 25.0 15.8 21.0 24.6 25.0 19.0 27.6 24.2 41.2
## [15] 43.4 38.6 42.0 42.6 34.4 31.0 18.2 39.2 42.0 33.4  8.4 43.4 38.2 27.0
## [29] 30.2 34.8 17.8 12.8 14.0 16.0 27.6 22.4 28.8 28.6
```

Next, these two data sets are combined as a single sample, and intercepts are set as covariance.

```
intercept1 = rep(b1, length(yValue1)); intercept1
```

```
##  [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [24] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
```

```
intercept2 = rep(b2, length(yValue2)); intercept2
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [36] 2 2 2
```

```
xValue = c(xValue1, xValue2)
yValue = c(yValue1, yValue2)
intercept = c(intercept1, intercept2)
intercept = as.factor(intercept); intercept
```

```
##  [1] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
## [24] 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 2  2  2 
## [47] 2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2 
## [70] 2  2  2  2  2  2  2  2  2  2  2  2 
## Levels: 2 17
```

```
regData = data.frame(yValue, xValue, intercept); regData
```

```
##    yValue xValue intercept
```

```
## 1     37.5   10.0          17
## 2     51.5   15.6          17
## 3     28.9    6.4          17
## 4     26.1    7.8          17
## 5     52.1   16.2          17
## 6     39.7    9.0          17
## 7     38.5    8.4          17
## 8     38.7    9.8          17
## 9     53.9   18.6          17
## 10    52.1   19.8          17
## 11    24.9    5.4          17
## 12    56.1   17.8          17
## 13    27.7    7.4          17
## 14    25.7    5.6          17
## 15    45.3   14.2          17
## 16    56.1   19.6          17
## 17    36.3   11.2          17
## 18    51.1   16.6          17
## 19    33.5    5.8          17
## 20    53.3   18.0          17
## 21    41.1   12.8          17
## 22    49.1   17.4          17
## 23    37.9   13.4          17
## 24    54.7   16.8          17
## 25    32.5   10.6          17
## 26    35.1    7.0          17
## 27    32.7    7.2          17
## 28    52.3   15.2          17
## 29    55.1   20.0          17
## 30    27.7    8.6          17
## 31    45.1   16.0          17
## 32    46.1   13.8          17
## 33    38.7   10.2          17
## 34    37.5   10.4          17
## 35    31.1    9.2          17
## 36    41.3   11.4          17
## 37    33.3    6.2          17
## 38    32.5    8.2          17
## 39    38.7   12.2          17
## 40    59.3   19.2          17
## 41    28.9    5.0          17
## 42    47.1   18.2          17
## 43    30.1    6.8          17
## 44    25.2    9.4           2
## 45    19.8    7.8           2
## 46    34.2   15.8           2
## 47    22.6    8.0           2
## 48    27.4   11.4           2
## 49    25.0   11.8           2
## 50    15.8    5.8           2
## 51    21.0    8.8           2
## 52    24.6   10.0           2
## 53    25.0    9.2           2
## 54    19.0    6.6           2
```

```
## 55   27.6   10.2         2
## 56   24.2   10.6         2
## 57   41.2   18.4         2
## 58   43.4   19.4         2
## 59   38.6   17.4         2
## 60   42.0   17.2         2
## 61   42.6   18.8         2
## 62   34.4   16.8         2
## 63   31.0   14.8         2
## 64   18.2    9.6         2
## 65   39.2   19.2         2
## 66   42.0   17.0         2
## 67   33.4   16.0         2
## 68    8.4    5.6         2
## 69   43.4   18.6         2
## 70   38.2   17.8         2
## 71   27.0   11.2         2
## 72   30.2   16.2         2
## 73   34.8   14.2         2
## 74   17.8    8.4         2
## 75   12.8    5.2         2
## 76   14.0    5.4         2
## 77   16.0    7.2         2
## 78   27.6   10.4         2
## 79   22.4    9.0         2
## 80   28.8   13.6         2
## 81   28.6   13.2         2
```

Hypothesis

H_0: There is no significant linear relationship between the yValue and the xValue. H_1: The yValue is linear to xValue.

Results

First, ANCOVA is performed. According to the results, I could draw two regerssion lines.

```r
reg = lm(yValue~xValue*intercept, data = regData)
summary(reg)
```

```
##
## Call:
## lm(formula = yValue ~ xValue * intercept, data = regData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2704 -2.2750  0.2072  2.4064  5.2194
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.19599    1.49632   2.136   0.0359 *
## xValue               2.02994    0.11465  17.705  < 2e-16 ***
## intercept17         13.66902    1.99911   6.838 1.69e-09 ***
```

```
## xValue:intercept17 -0.04207    0.15368  -0.274   0.7850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.156 on 77 degrees of freedom
## Multiple R-squared:  0.9299, Adjusted R-squared:  0.9271
## F-statistic: 340.3 on 3 and 77 DF,  p-value: < 2.2e-16
```

```
plot(xValue, yValue, col=c('blue', 'red')[as.numeric(intercept)])
abline(lm(yValue[intercept==b1]~xValue[intercept==b1]), lty=1, col='blue')
abline(lm(yValue[intercept==b2]~xValue[intercept==b2]), lty=1, col='red')
```

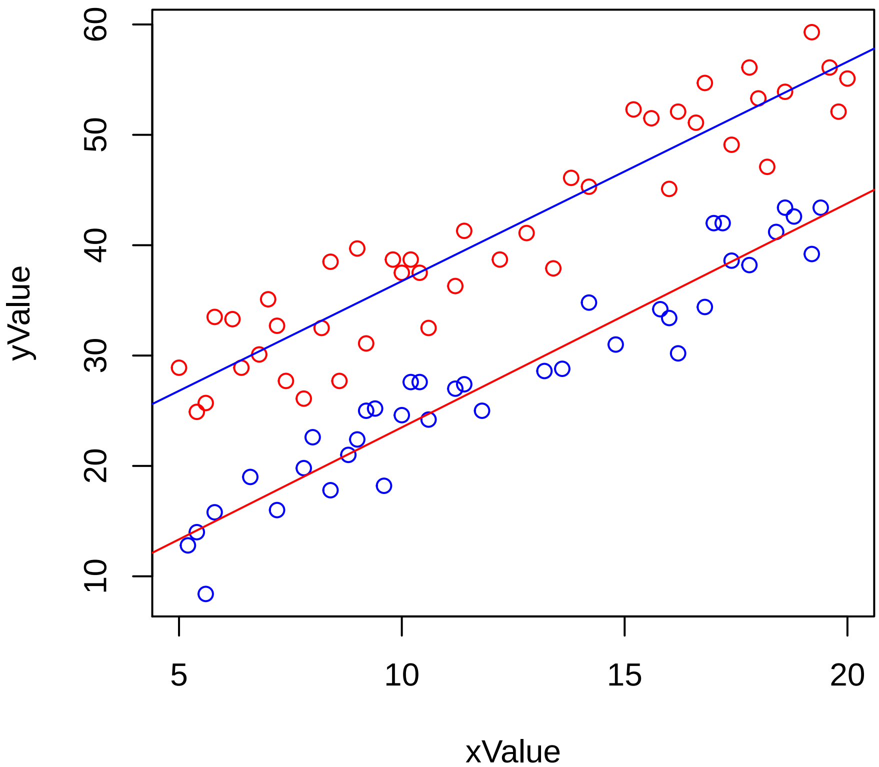

Figure 1. Fitted data after ANCOVA

Automated model selection is also performed.

```
step(reg)
```

```
## Start:  AIC=190.09
## yValue ~ xValue * intercept
##
##                    Df Sum of Sq    RSS    AIC
## - xValue:intercept  1   0.74646 767.74 188.17
## <none>                          766.99 190.09
##
## Step:  AIC=188.17
## yValue ~ xValue + intercept
##
##             Df Sum of Sq    RSS    AIC
## <none>                    767.7 188.17
## - intercept  1    3490.4 4258.1 324.93
## - xValue     1    6879.9 7647.6 372.36
```

```
##
## Call:
## lm(formula = yValue ~ xValue + intercept, data = regData)
##
## Coefficients:
## (Intercept)       xValue  intercept17
##       3.483        2.007       13.157
```

Finally, residules after fitting is plotted to test the normality of the samples.

```r
plot(residuals(reg))
```

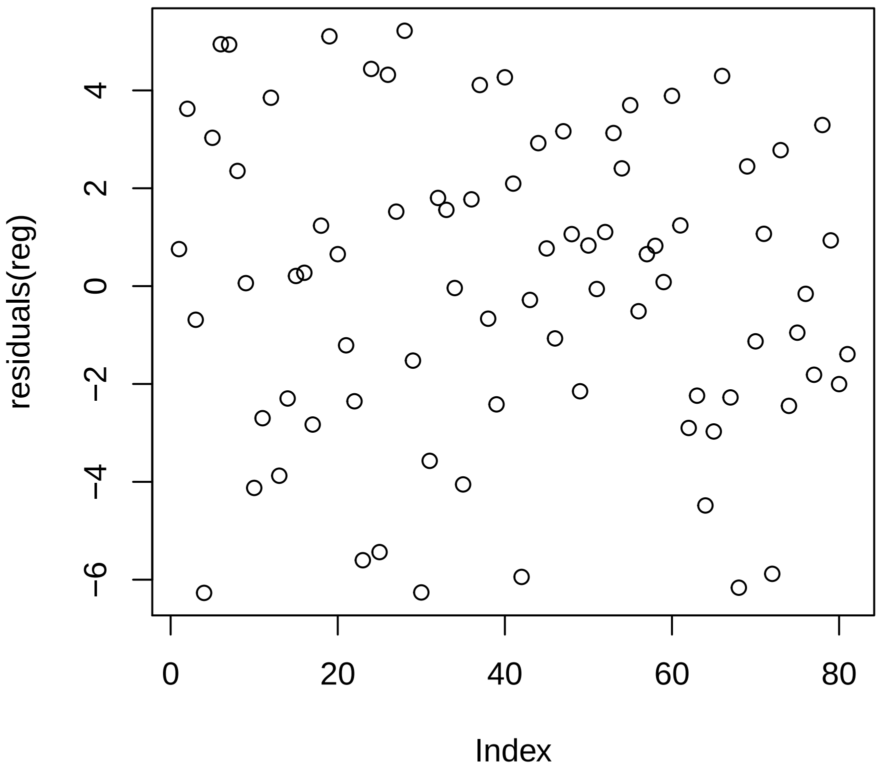

Figure 2. Check homogeneity of variance