# Regression Analysis Examples in R

## Simple Linear Regression

### Example. Airfreight Data

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Shipment Route ($x$) | 1 | 0 | 2 | 0 | 3 | 1 | 0 | 1 | 2 | 0 |
| Airfreight Breakage ($y$) | 16 | 9 | 17 | 12 | 22 | 13 | 8 | 15 | 19 | 11 |

a) Compute the ANOVA table
b) Compute the confidence intervals for the parameters
c) Compute the confidence interval on the average (mean) response when $X = 1$.

### Solution.

#### Part a.

We can compute the anova table manually as follows,

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2 = 20 - \frac{1}{10}(100) = 10$$

$$S_{xy} = \sum_{i=1}^{n} y_i(x_i - \bar{x}) = \sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i = 182 - \frac{1}{10}(10)(142) = 40$$

Therefore,

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{40}{10} = 4$$

Then, $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$, so

$$\hat{\beta}_0 = \frac{1}{10}(142) - 4 \cdot \frac{1}{10}(10) = 10.2$$

This gives us our linear model

$$\hat{y} = 10.2 + 4x$$

The sum of squares for regression is

$$SSR = \hat{\beta}_1^2 \sum_{i=1}^{n}(x_i - \bar{x})^2 = \hat{\beta}_1^2 S_{xx} = 16 \cdot 10 = 160$$

The total sum of squares is

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 = 2194 - 10(14.2)^2 = 177.6$$

Then, the residual sum of squares is

$$SSE = SST - SSR = 177.6 - 160 = 17.6$$

Now we can construct the anova table

| Source | Sum of Squares | DF | MS=SS/df | F = MSR/MSE |
|---|---|---|---|---|
| Regression | 160 | 1 | 160 | 72.727 |
| Error | 17.6 | 8 | 2.2 | |
| Total | 177.6 | | | |

We conclude tbat the regression is highly significant since the $F$ value is very large. We can also do this in R

```r
x <- c(1,0,2,0,3,1,0,1,2,0)
y <- c(16,9,17,12,22,13,8,15,19,11)
model <- lm(formula = y ~ x)
print(model)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##        10.2          4.0
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1  160.0   160.0  72.727 2.749e-05 ***
## Residuals  8   17.6     2.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that we get the same results and reach the same conclusion.

**Part b.**

We can construct confidence intervals, first we need to compute $se(\hat{\beta}_1)$ and $se(\hat{\beta}_0)$.

$$se^2(\hat{\beta}_0) = MSE\left(\frac{1}{n} + \frac{\bar{x}}{S_{xx}}\right) = 2.2\left(\frac{1}{10} + \frac{1}{10}\right) = 0.44 \implies se(\hat{\beta}_0) = \sqrt{0.44} = 0.6633$$

$$se^2(\hat{\beta}_1) = \frac{MSE}{S_{xx}} = \frac{2.2}{10} = 0.22 \implies se(\hat{\beta}_1) = \sqrt{0.22} = 0.490$$

Then, we have to compute $t_{\alpha/2,n-2} = t_{0.025,8}$, we either use a $t$ look up table or in R,

```r
qt(0.025, 8, lower.tail=FALSE)
```

```
## [1] 2.306004
```

Thus, our confidence intervals are

$$\hat{\beta}_0 - t_{\alpha/2,n-2}se(\hat{\beta}_0) \leq \hat{\beta}_0 \leq \hat{\beta}_0 + t_{\alpha/2,n-2}se(\hat{\beta}_0) \rightarrow 10.2 \pm 2.306(0.6633) = (8.6704, 11.7296)$$

$$\hat{\beta}_1 - t_{\alpha/2,n-2}se(\hat{\beta}_1) \leq \hat{\beta}_1 \leq \hat{\beta}_1 + t_{\alpha/2,n-2}se(\hat{\beta}_1) \rightarrow 4 \pm 2.306(0.490) = (2.9392, 5.0608)$$

We can compute these confidence intervals in R as well

```r
confint(model, level=0.95)
```

```
##                2.5 %    97.5 %
## (Intercept) 8.670370 11.729630
## x           2.918388  5.081612
```

### Part c.

We want to compute first $E(y|x_0)$, where $x_0 = 1$. An unbiased estimator for $E(y|x_0)$ is

$$\widehat{E(y|x_0)} = \hat{\mu}_{y|x_0} = \hat{\beta}_0 + \hat{\beta}_1 x_0 = 10.2 + 4(1) = 14.2$$

Then, the confidence interval is

$$\left[\hat{\mu}_{y|x_0} \pm t_{\alpha/2,n-2}\sqrt{MSE\left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}\right] = \left[14.2 \pm 2.306\sqrt{2.2\left(\frac{1}{10} + \frac{(1-1)^2}{S_{xx}}\right)}\right] = (13.11839, 15.28161)$$

We can do this in R with

```r
predict(model, newdata = data.frame(x=1), interval = 'confidence', level=0.95)
```

```
##    fit      lwr      upr
## 1 14.2 13.11839 15.28161
```