4.3 Lack-Of-Fit Tests

So far we have discussed how to address lack of constant variance, normality, and uncorrelated errors assumptions. In this section, we discuss how to check the *linearity* assumption by introducing **lack-of-fit** tests.

Recall that under the usual assumptions for the error terms we have that

$$\hat{\sigma}^2 = \frac{RSS}{n-p} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|^2}{n-p} \sim \sigma^2 \frac{\chi^2_{n-p}}{n-p}.$$

If the model is correct, then $\hat{\sigma}^2$ is an **unbiased** estimate of σ^2.

4.3.1 LoF Test: σ^2 Known

In the very special case where we knew σ^2, we could construct a test based on the ratio

$$\frac{\hat{\sigma}^2}{\sigma^2},$$

i.e. a measure of lack-of-fit.

The hypothesis to test is

$$\begin{cases} H_0 : \text{There is no lack of fit.} \\ H_\alpha : \text{There is lack of fit.} \end{cases}$$

and the corresponding test statistic formulates as

$$\frac{\hat{\sigma}^2}{\sigma^2} = \frac{RSS/(n-p)}{\sigma^2} \sim \frac{\chi^2_{n-p}}{n-p}.$$

This test statistic takes *large* values when $\hat{\sigma}^2$ is larger than σ^2. We have *"Lack-of-Fit"* when the error variance is *large* related to the value of σ^2.

Lack-of-Fit

$$\begin{cases} H_0 : \text{There is no lack of fit.} \\ H_\alpha : \text{There is lack of fit.} \end{cases}$$

We conclude that there is lack-of-fit (i.e. *Reject* H_0), if:

$$(n-p)\,\frac{\hat{\sigma}^2}{\sigma^2} \geq \chi^2_{n-p}(1-\alpha)$$

`strongx` Example

In this example, all individual variances have been accounted for using *weighted least squares*, so we have $\sigma^2 = 1$.

Then, under H_0, our test is based on

$$(n-p)\,\hat{\sigma}^2 \sim \chi^2_{n-p}$$

```
strong.weights = lm(crossx ~ energy, strongx, weights=1/sd^2)
1 - pchisq(summary(strong.weights)$sig^2*8, 8) # Assume sigma^2=1
```

```
## [1] 0.005004345
```

Since the p-value < 0.05, we reject the null hypothesis and conclude that there is a lack-of-fit. This might be the case even with a high value of R^2.

4.3.2 LoF Test: σ^2 Unknown

If σ^2 is unknown, a general approach is to compare an estimate of σ^2 based on a *much bigger/general model*. If we can derive the distribution (under H_0) of

$$\hat{\sigma}^2_{LinearModel} / \hat{\sigma}^2_{BigModel},$$

then we *reduce* this problem to a two model comparison test problem.

The null hypothesis is the current model:

$$H_0 : \mathbb{E}(y_i) = \mathbf{x}_i^\top \beta, \quad i = 1, 2, \ldots, n, \quad \text{for some vector } \beta$$

The more *general* model is assumed under the alternative hypothesis:

$$H_\alpha : \mathbb{E}(y_i) = f(\mathbf{x}_i), \quad i = 1, 2, \ldots, n, \text{ for some function} f$$

We can estimate σ^2 for the big model in H_α, if there is some **replication** in the data, i.e., there are multiple observations (replicates) for some (at least) of the same $\mathbf{x}_i$ values. Schematically we can represent these replicates as:

$$(\mathbf{x}_i, y_{i1}, y_{i2}, \ldots, y_{in_i}), \quad i = 1 : m, \quad n = \sum_i n_i$$

Lack-Of-Fit Test

Under the null hypothesis H_0:

$$y_{ij} = \mathbf{x}_i^\top \beta + \varepsilon_{ij}$$

for some β, and $\varepsilon_{ij} \sim^{iid} \mathcal{N}(0, \sigma^2)$. In this case the corresponding RSS_0 with $df = n - p$.

Under the alternative big-model hypothesis H_α:

$$y_{ij} = f(\mathbf{x}_i) + \varepsilon_{ij}$$

for some (smooth) function f, and $\varepsilon_{ij} \sim^{iid} \mathcal{N}(0, \sigma^2)$. In this case the corresponding RSS_α with $df = n - m = \sum_i (n_i - 1)$, where

$$RSS_\alpha = \sum_{i=1}^{m} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i.})^2$$

All of the degrees of freedom for RSS_α come from the replications. Therefore, *with replication* we can do an F test for lack-of-fit:

$$F = \frac{(RSS_0 - RSS_a)/(m - p)}{RSS_a/(n - m)} \sim F_{m-p, n-m}$$

Corrosion Data Set

Data consist of thirteen specimens of 90/10 Cu-Ni alloys with varying iron content in percent. The specimens were submerged in sea water for 60 days and the weight loss due to corrosion was recorded in units of milligrams per square decimeter per day.

- Fe: Iron content in percent loss
- loss: Weight loss in mg per square decimeter per day

```
library(faraway)
data("corrosion")
corrosion[order(corrosion$Fe),]
```

```
##      Fe  loss
## 1  0.01 127.6
## 6  0.01 130.1
## 11 0.01 128.0
## 2  0.48 124.0
## 7  0.48 122.0
## 3  0.71 110.8
## 9  0.71 113.1
## 4  0.95 103.9
## 5  1.19 101.5
## 8  1.44  92.3
## 12 1.44  91.4
## 10 1.96  83.7
## 13 1.96  86.2
```

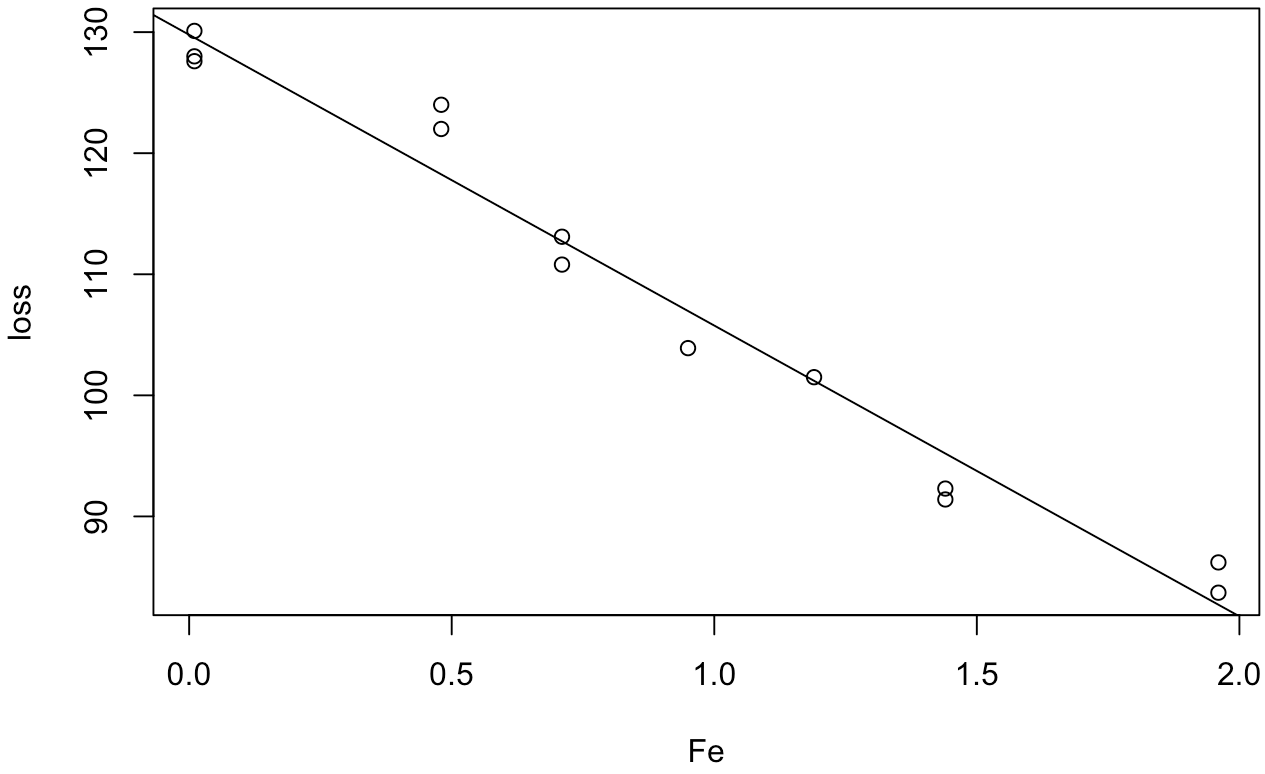
For a given value of iron content (x_i), we have several observations of weight loss (y_{ij}). We start by fitting a SLR model

```
corrosion.slr = lm(loss ~ Fe, data=corrosion)
summary(corrosion.slr)
```

```
##
## Call:
## lm(formula = loss ~ Fe, data = corrosion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7980 -1.9464  0.2971  0.9924  5.7429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  129.787      1.403   92.52  < 2e-16 ***
## Fe           -24.020      1.280  -18.77 1.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.058 on 11 degrees of freedom
## Multiple R-squared:  0.9697, Adjusted R-squared:  0.967
## F-statistic: 352.3 on 1 and 11 DF,  p-value: 1.055e-09
```

The scatterplot of the raw data and the corresponding regression line is shown below:

```
plot(loss~Fe, data=corrosion)
abline(coef(corrosion.slr))
```




```
corrosion.larger=lm(loss ~ factor(Fe), data=corrosion);
cbind(corrosion, corrosion.larger$fitted)[order(corrosion$Fe),]
```

```
##      Fe  loss corrosion.larger$fitted
## 1  0.01 127.6                128.5667
## 6  0.01 130.1                128.5667
## 11 0.01 128.0                128.5667
## 2  0.48 124.0                123.0000
## 7  0.48 122.0                123.0000
## 3  0.71 110.8                111.9500
## 9  0.71 113.1                111.9500
## 4  0.95 103.9                103.9000
## 5  1.19 101.5                101.5000
## 8  1.44  92.3                 91.8500
## 12 1.44  91.4                 91.8500
## 10 1.96  83.7                 84.9500
## 13 1.96  86.2                 84.9500
```

```
summary(corrosion.larger)
```

```
##
## Call:
## lm(formula = loss ~ factor(Fe), data = corrosion)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2500 -0.9667  0.0000  1.0000  1.5333
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     128.567      0.809 158.914 4.19e-12 ***
## factor(Fe)0.48   -5.567      1.279  -4.352  0.00481 **
## factor(Fe)0.71  -16.617      1.279 -12.990 1.28e-05 ***
## factor(Fe)0.95  -24.667      1.618 -15.245 5.03e-06 ***
## factor(Fe)1.19  -27.067      1.618 -16.728 2.91e-06 ***
## factor(Fe)1.44  -36.717      1.279 -28.703 1.18e-07 ***
## factor(Fe)1.96  -43.617      1.279 -34.097 4.24e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.401 on 6 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9931
## F-statistic: 287.3 on 6 and 6 DF,  p-value: 4.152e-07
```

```
anova(corrosion.slr, corrosion.larger)
```

```
## Analysis of Variance Table
##
## Model 1: loss ~ Fe
## Model 2: loss ~ factor(Fe)
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     11 102.850
## 2      6  11.782  5    91.069 9.2756 0.008623 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
1-pf(9.2756,5,6) #There is lack of fit
```

```
## [1] 0.008622884
```

The model under H_0 is compared with a more general model in where each level of X is considered as a factor.Since the p-value < 0.05 we have Lack of Fit. The model under H_0 is not adequate for this data set.