Printout

Saturday, April 1, 2023 9:29 AM

Section 1

MANE 6313

Subsection 1

Week 12, Module C

Student Learning Outcome

- Select an appropriate experimental design with one or more factors,
- Select an appropriate model with one or more factors,
- Evaluate statistical analyses of experimental designs,
- Assess the model adequacy of any experimental design, and
- Interpret model results.

Module Learning Outcome

Explaining inference for linear regression models.

Test for Significance of Regression

$H_0: \beta_1 = \beta_2 = \cdots = \beta_k$

$H_1:$ at least one $\beta_i \neq 0$

Tests on Individual Regression Coefficients

- We can test $H_0 : \beta_j = 0$ vs. $H_a : \beta_j \neq 0$

$$\frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 C_{jj}}} \sim t_{n-k-1}$$

where C_{jj} is the (jj)th element of $(\mathbf{X}'\mathbf{X})^{-1}$

- or equivalently

$$\frac{\hat{\beta}_j}{se(\hat{\beta}_j)} \sim t_{n-k-1}$$

- If $H_0 : \beta_j = 0$ is not rejected, we can remove the variable x_j from the model

Example Problem 12.8

```r
```{r}
ex12_8.model <- lm(GrainRadius~PowderTemp+Extrusion+DieTemp,data=ex12_8.df)
summary(ex12_8.model)
```
```

```
Call:
lm(formula = GrainRadius ~ PowderTemp + Extrusion + DieTemp,
    data = ex12_8.df)

Residuals:
    1     2     3     4     5     6     7     8
-0.75  5.25  1.75 -2.25 -6.25 -2.25  1.25  3.25

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -284.50000   30.77729  -9.244 0.000761 ***
PowderTemp     0.12500    0.08501   1.470 0.215398
Extrusion      2.45833    0.28336   8.676 0.000972 ***
DieTemp        1.45000    0.11335  12.793 0.000215 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.809 on 4 degrees of freedom
```

C.I. on the Individual Regression Coefficients

- Straightforward to construct

$$\frac{\hat{\beta}_j - \beta}{\sqrt{\hat{\sigma}^2 C_{jj}}} \sim t_{n-p} \quad j = 0, 1, 2, \ldots, k$$

- A $100(1-\alpha)\%$ confidence interval for β_j is

$$\hat{\beta}_j - t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 C_{jj}} \leq \beta_j \leq \hat{\beta}_j + t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 C_{jj}}$$

- or equivalently

$$\hat{\beta}_j - t_{\alpha/2,n-p}se(\hat{\beta}_j) \leq \beta_j \leq \hat{\beta}_j + t_{\alpha/2,n-p}se(\hat{\beta}_j)$$

R: confint() Function

confint: Confidence Intervals for Model Parameters

Description

Computes confidence intervals for one or more parameters in a fitted model. There is a default and a method for objects inheriting from class `"lm"`.

Usage

```
confint(object, parm, level = 0.95, …)
```

Arguments

`object`	a fitted model object.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required.
…	additional argument(s) for methods.

Figure 2: confint() Documentation

Example 12.8 Confidence Interval on Parameters

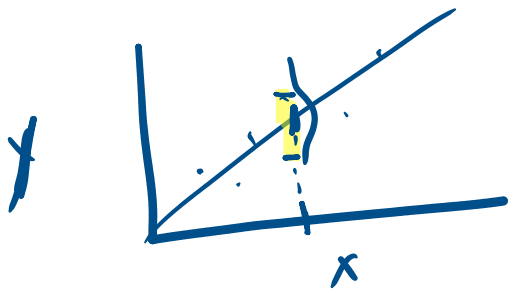
```r
confint(ex12_8.model,level=0.95)
```

95% confidence interval

	2.5 %	97.5 %
(Intercept)	-369.9514640	-199.0485360
PowderTemp	-0.1110234	0.3610234
Extrusion	1.6715888	3.2450778
DieTemp	1.1353022	1.7646978

Interval containing 0. If interval contains 0, equivalent to failing to Reject

Figure 3: Ex12.8 C.I. on Parameters



C.I. on the Mean response

- We can find a confidence interval on the mean response at a point $x_0' = [1, x_{01}, x_{02}, \ldots, x_{x0k}]$.
- Note that

$$\mu_{y|\mathbf{x_0}} = \beta_0 + \beta_1 x_{01} + \beta_2 x_{02} + \cdots + \beta_k x_{0k}$$
$$\hat{y}(\mathbf{x_0}) = \mathbf{X_0'}\widehat{\beta}$$

- The variance of $\hat{y}(\mathbf{x_0})$ is

$$V\left[\hat{y}(\mathbf{x_0})\right] = \sigma^2 \mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0}$$

- A $100(1-\alpha)\%$ confidence interval for the mean response is

$$\hat{y}(\mathbf{x_0}) - t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 \mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0}}$$
$$\leq \mu_{y|x_0} \leq \hat{y}(\mathbf{x_0}) + t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 \mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0}}$$

Prediction Intervals

- We can use the regression equation to predict values at points other than those in the design matrix. In general, we only want to interpolate; not extrapolate
- Consider the point $x_0' = [1, x_{01}, x_{02}, \ldots, x_{x0k}]$. A point estimate for y is

$$\hat{y}(\mathbf{x_0}) = \mathbf{x_0'}\widehat{\beta}$$

- A $100(1-\alpha)\%$ prediction interval for this observation is

$$\hat{y}(\mathbf{x_0}) - t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2(1+\mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0})} \leq y_0 \leq \hat{y}(\mathbf{x_0}) + t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2(1+\mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0})}$$

- Most packages will perform this function for you.

R: predict.lm() Function

Source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/predict.lm

stats (version 3.6.2)

predict.lm: Predict method for Linear Model Fits

Description

Predicted values based on linear model object.

Usage

```
# S3 method for lm
predict(object, newdata, se.fit = FALSE, scale = NULL, df = Inf,
        interval = c("none", "confidence", "prediction"),
        level = 0.95, type = c("response", "terms"),
        terms = NULL, na.action = na.pass,
        pred.var = res.var/weights, weights = 1, …)
```

Figure 4: predict.lm() Documentation

Example 12.8 - Confidence Interval

```r
new.df <- data.frame(170,18,235)
names(new.df) <- c('PowderTemp','Extrusion','DieTemp')
predict.lm(ex12_8.model,new.df,interval="confidence",level = 0.98)
```

```
      fit      lwr      upr
1 121.75 115.3795 128.1205
```

Figure 5: predict.lm() Confidence Interval

Example 12.8 - Prediction Interval

```r
new.df <- data.frame(170,18,235)
names(new.df) <- c('PowderTemp','Extrusion','DieTemp')
predict.lm(ex12_8.model,new.df,interval="prediction",level = 0.98)
```

```
      fit      lwr      upr
1 121.75 102.6385 140.8615
```

Figure 6: predict.lm() Prediction Interval