22S:164:Lab6

Oct.4, 2007

1. Weighted Least Squares

- Weighted least squares estimates are most easily obtained using the `weights` for the `lm` command. We use the `physics` data as our example for the WLS regression.

```
>library(alr3)
>data(physics);attach(physics)
>m1 <- lm(y~x,weights=1/SD^2)
>summary(m1); anova(m1)
```

- To return the correct residuals for WLS, we can use either `>residuals(m1,type="pearson")` or `>weighted.residuals(m1)`. For OLS all weights equal one, so the Pearson and ordinary residuals are identical.
- The predict function also works correctly for getting predictions and standard errors of fitted values, but it doesn't always give the right answer for prediction intervals. The R assumes that the variance of the future observation is σ^2 rather than σ^2 / w_*, where w_* is the weight for the future value.

2. Calculate WLS Statistics Using Matrix Commands

Next, we will show how to calculate statistics for the `physics` data frame by using matrix commands in R.

- First of all, create a predictor matrix **X,** a weight matrix **W** and a vector **y** as follows:

```
> X <- cbind(1,x); X<- as.matrix(X)
> Y <-as.matrix(y)
> w <-diag(1/SD^2)
```

- Now use R commands to compute $\mathbf{X}'W\mathbf{X}$, $\mathbf{X}'WY$ and $\hat{\boldsymbol{\beta}}_w$

```
>XwX <- t(X)%*%w%*%X
>XwY <- t(X)%*%w%*%Y
>betahat <- solve(XwX,XwY)
```

3. Testing for Lack of Fit

- To fit the `physics` data in a quadratic model, use the following commands:

```
>m2 <-lm(y~x+I(x^2),weights=1/SD^2)
>summary(m2)
>anova(m2)
```

- Here are the commands that generate both linear and quadratic plots together in one graph:

```
> plot(x,y)
> abline(m1)
> a<-seq(0.05,0.35,length=50)
> lines(a,predict(m2,newdata=data.frame(x=a)),lty=2)
```

- In order to get the test of lack of fit based on replication, the easiest way is to add `factor(x)` to the mean function and compare the `anova` of two models. Look at the following model:

```
> x <- rep(c(1:4),c(3,1,2,4)) #see what this command does
> y <- c(2.55,2.75,2.57,2.40,4.19,4.7,3.81,4.87,2.93,4.52)
> m1 <- lm(y~x)
> m2 <- lm(y~factor(x))
> anova(m1,m2)
```

- Alternatively, you can use the build-in function `pure.error.anova` in library `alr3` to do this trick:

```
> pure.error.anova(m1)
```