22s:164:Lab9

Nov 1, 2007

1. Regression Diagnostics: Residuals

- To obtain the hat-matrix is very easy in R. the command `hatvalues` works for both OLS and WLS. For `physics` data, recall that to fit a weighted least square model, we have to specify the weights and use Pearson's residuals as the correct residuals.

```
> library(alr3)
> data(physics);attach(physics)
> m1 <- lm(y~x,weights=1/SD^2)
> residuals(m1, type="pearson")    #residuals for WLS
> hatvalues(m1)                    #Leverages
> rstandard(m1)                    #standardized residuals
> rstudent(m1)                     #studentized residuals
> cooks.distance(m1)               #Cook's distance
```

To see more commands for diagnostics, type `> help(influence.measures)`.

2. Testing for Curvature

- The command residual.plots implements the curvature tests as well as the residual plots. For the fuel2001 data, the residual plots, along with a fitted quadratic curve, and the curvature tests, are obtained using the following commands:

```
# again, we need to modify the data set first:
> data(fuel2001);attach(fuel2001)
> Dlic<-1000*Drivers/Pop; Fuel<-1000*FuelC/Pop
> Income<-Income/1000; logMiles<-log(Miles,2)
> f<-data.frame(Tax,Dlic,Income,logMiles,Fuel)
# fit an OLS model with all predictors
> full.lm <-lm(Fuel~Tax+Dlic+Income+logMiles,data=f)
> residual.plots(full.lm)
```

- If you have any factors in your formula, boxplots of residuals will be drawn for the factors, but there is no curvature test for the factor. Interactions are skipped entirely.

3. Nonconstant Variance

- We will use the snow geese data to illustrate the test for heteroscedasticity. Type the following commands:

```
> data(snowgeese)
> m1 <- lm(photo~obs1,snowgeese)
> sig2<-sum(residuals(m1,type="pearson")^2)/length(snowgeese$obs1)
> U<-residuals(m1)^2/sig2
> m2 <- update(m1,U~.)
> anova(m2)
```

The model with term ob1 is fit. The score variable U is computed, and then m2 is the regression of U on the terms in the original model. The score test is 1/2 the sum of squares for regression.

- Alternatively, there is a command that automates computing the score test for nonconstant variance. The command called `ncv.test` is part of the `car` package.

```
> library(car)
> ncv.test(m1)
```

- Let's look at another example with more variance functions in the `sniffer` data. The `ncv.test` gives the results of several nonconstant variance score tests, each computed using a different choice of Z

```
> data(sniffer)
> attach(sniffer)
> m1 <- lm(Y~TankTemp+GasTemp+TankPres+GasPres)
> residual.plots(m1)
> ncv.test(m1,~TankTemp)
> ncv.test(m1,~TankTemp+GasPres)
> ncv.test(m1,~TankTemp+GasTemp+TankPres+GasPres)
> ncv.test(m1)
```

4. Graphs For Model Assessment

- The alr3 package provides two commands for drawing marginal model plots: `mmp` draws one marginal model plot; `mmps` draws many such plots. The command `mmp` uses the loess smoother, with default smoothing parameter 2/3. Now, look at the `ufcwc` example.

```
> data(ufcwc)
> c1<-lm(Height~Dbh,ufcwc)
> mmp(c1,ufcwc$Dbh)
> mmp(c1)
> mmp(c1,sd=TRUE)
```

The first argument of `mmp` is the name of the regression model, which is required. The second argument is the quantity to put on the horizontal axis. The default one is the fitted value of the model. If the argument `sd` is `TRUE`, the standard deviation smooths will be added to the plot.

- To draw marginal model plots versus each predictor and versus fitted values, use `mmps`. The following example uses `UN2` data.

```
> data(UN2)
> attach(UN2)
> m1 <- lm(logFertility~logPPgdp+Purban)
> mmps(m1)
```

By adding a quadratic term into the model, the mean function matches the data well. The marginal model plots are obtained using following commands:

```
> m2 <- lm(logFertility~logPPgdp+Purban+I(Purban^2))
> mmps(m2)
```