Regression Diagnostics in R/S-Plus

Example: Stack Loss Data

```r
> stackloss <- read.table("http://www.stat.duke.edu/courses/Spring03/sta244/Data/stackloss", header=T)

# Always plot the data!!!
> postscript("stackloss-pair.ps")
> pairs(stackloss)
> dev.off()
```

```
StackLoss

AirFlow

WaterTemp

AcidConc
```

```r
# Fit the linear model with all variables, no transformations
> lm1 <- lm(StackLoss ~ ., qr=T, data=stackloss)

# use qr = T as an option in order to obtain additional diagnostics later.

> par(mfrow=c(2,2))
> plot(lm1)
> postscript("stackloss-residuals.ps")
> dev.off()

Anything alarming?
```
```r
> summary(lm1)
Call:
  lm(formula = StackLoss ~ ., data = stackloss, qr = T)

Residuals:
  Min 1Q Median 3Q Max
-7.2377 -1.7117 -0.4551 2.3614 5.6978

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -39.9197  11.8960  -3.356  0.00375 **
  AirFlow     0.7156   0.1349   5.307  5.8e-05 ***
WaterTemp   1.2953   0.3680   3.520  0.00263 **
     AcidConc -0.1521   0.1563  -0.973  0.34405

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.243 on 17 degrees of freedom
Multiple R-Squared: 0.9136, Adjusted R-squared: 0.8983
F-statistic: 59.9 on 3 and 17 DF, p-value: 3.016e-09
```
Diagnostics

The function `ls.diag` is used to obtain many standard regression diagnostics. It was originally used with an argument that was the output of the function `lsfit`, but if you use `qr=T` in the `lm` command, you can use `ls.diag()` with `lm`'s output too. The output is a list with the following numeric components:

- `std.dev`: The standard deviation of the errors, an estimate of sigma.
- `hat`: diagonal entries $h_{ii}$ of the hat matrix $H$
- `std.res`: standardized residuals
- `stud.res`: studentized residuals
- `cooks`: Cook's distances
- `dfts`: DFITS statistics
- `correlation`: correlation matrix
- `std.err`: standard errors of the regression coefficients
- `cov.scaled`: Scaled covariance matrix of the coefficients
- `cov.unscaled`: Unscaled covariance matrix of the coefficients

Back to the example:

```r
> stack.diag <- ls.diag(lm1)
> names(stack.diag)
[1] "std.dev" "hat" "std.res" "stud.res" "cooks"
[6] "dfits" "correlation" "std.err" "cov.scaled" "cov.unscaled"
```

```r
> par(mfrow=c(2,2))
> plot(abs(stack.diag$std.res), ylab="|Internally Studentized Residuals|", xlab="Case", type="h")
> plot(stack.diag$hat, ylab="Leverage", xlab="Case", type="h")
> plot(abs(stack.diag$stud.res), ylab="|Externally Studentized Residuals|", xlab="Case", type="h")
> plot(stack.diag$cooks, ylab="Cook's Distance", type="h", xlab="Case")
```

```r
> 2*(1- pt(max(abs(stack.diag$stud.res)), 16)) #p-value for largest stud residual
[1] 0.00423804
> .05/21 # Bonferonni adjustment for alpha
[1] 0.002380952
> max(abs(stack.diag$stud.res))
[1] 3.330493
> qt(1 - .025/21, 16) # critical value for rejection region with Bonferonni adjustment
[1] 3.603616
```

What do we conclude from this?