Model results and diagnostics

There are many functions available to produce predicted values and diagnostics. For additional commands not listed here, see help (influence. measures)).

Predicted values

```
mod1 = lm(y ~ x, data=ds)
predicted.varname = predict(mod1)
```

Residuals

```
mod1 = lm(y ~ x, data=ds)
residual.varname = residuals(mod1)
```

Standardized and Studentized residuals

```
mod1 = lm(y ~ x, data=ds)
standardized.resid.varname = rstandard(mod1)
studentized.resid.varname = rstudent(mod1)
```

Leverage

```
mod1 = lm(y ~ x, data=ds)
leverage.varname = hatvalues(mod1)
```

Cook's distance

```
mod1 = lm(y ~ x, data=ds)
cookd.varname = cooks.distance(mod1)
```

DFFITs

```
mod1 = lm(y ~ x, data=ds)
dffits.varname = dffits(mod1)
```

Diagnostic plots

```
mod1 = lm(y ~ x, data=ds)
par(mfrow=c(2,2)) # display 2 x 2 matrix of graphs
plot(mod1)
```

Heteroscedasticity tests

```
library(lmtest)
bptest(y ~ x1 + ... + xk, data=ds)
```

Parameter estimates

```
mod1 = lm(y ~ x, data=ds)
coeff.mod1 =coef(mod1)
```

Standardized regression coefficients

```
library(QuantPsyc)
mod1 = lm(y ~ x)
lm.beta(mod1)
```

Confidence interval for parameter estimates

```
mod1 = lm(y ~ x, data=ds)
confint(mod1)
```

Prediction limits

```
mod1 = lm(y ~ ..., data=ds)
pred.w.lowlim = predict(mod1, interval="prediction")[,2]
```

Covariance matrix of parameter estimates

```
mod1 = lm(y ~ x, data=ds)
vcov(mod1)
```

Correlation matrix of parameter estimates

```
mod1 = lm(y ~ x, data=ds)
mod1.cov = vcov(mod1)
mod1.cor = cov2cor(mod1.cov)
```