

Command	Explanation	
<code>lm()</code>	runs a linear regression	
<code>predict()</code>	calculate predicted value	
<code>residuals()</code>	calculates residuals of regression	
<code>linearHypothesis()</code>	tests for joint significance	car
<code>resettest()</code>	performs a RESET test	lmtest
<code>jarque.bera.test()</code>	performs a Jarque-Bera test	tseries

Examples

```
regression <- lm(y ~ x)
```

OLS estimation, regresses vector `y` on `x`, saves results as `regression`.

```
regression <- lm(y ~ x, data = df)
```

OLS estimation, regresses variable `y` on `x` from data frame `df`, saves results as `regression`.

```
regression <- lm(y ~ x + z, data = df)
```

OLS estimation, regresses variable `y` on `x` and `z` from data frame `df`, saves results as `regression`.

```
prednums <- data.frame(x = 2, z = 4)
```

```
predict(regression, prednums)
```

Plugs $x = 2$ and $z = 4$ into the estimated regression, generating predicted value for y .

```
residuals(regression)[7]
```

Shows the residual of the 7th observation for model `regression`.

```
df$logx = log(df$x)
```

Takes the logarithm of variable `x` from data frame `df` and saves it back into the data frame as `logx`.

```
Hnull <- c("sibs=0", "brthord=0")
```

```
linearHypothesis(regur, Hnull)
```

Tests the joint significance of regressors *sibs* and *brthord* in the (unrestricted) model `regur`.

```
resettest(reg)
```

Uses RESET test to see if model `reg` is misspecified by using squared and cubed fitted values.

```
jarque.bera.test(reg$residuals)
```

Uses Jarque-Bera test to see if model `reg` has normally distributed residuals.