Transformations and Polynomial Regression

One of the first steps in the construction of a regression model is to hypothesize the form of the regression function. We can dramatically expand the scope of our model by including specially constructed explanatory variables. These include indicator variables, interaction terms, transformed variables, and higher order terms. In this tutorial we discuss the inclusion of transformed variables and higher order terms in your regression models.

A. Transforming Data

When fitting a linear regression model one assumes that there is a linear relationship between the response variable and each of the explanatory variables. However, in many situations there may instead be a non-linear relationship between the variables. This can sometimes be remedied by applying a suitable transformation to some (or all) of the variables, such as power transformations or logarithms. In addition, transformations can be used to correct violations of model assumptions such as constant error variance and normality.

We apply transformations to the original data prior to performing regression. This is often sufficient to make linear regression models appropriate for the transformed data.

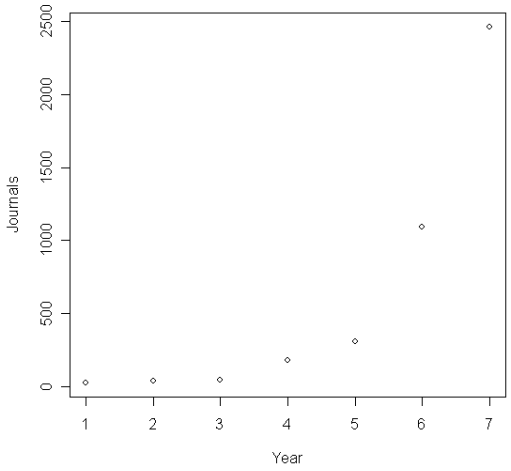
Ex. Data was collected on the number of academic journals published on the Internet during the period 1991-1997.

Year	1	2	3	4	5	6	7
Journals	27	36	45	181	306	1093	2459

Note: The years are re-expressed as (year-1990).

Begin by reading in the data and making a scatter plot of year and number of journals.

```
> Year = seq(1,7)
> Journals = c(27,36,45,181,306,1093,2459)
> Internet = data.frame(Year,Journals)
> plot(Year,Journals)
```



The plot indicates that there is a clear nonlinear relationship between the number of journals and year. Because of the apparent exponential growth in journals, it appears that taking the logarithm of number of journals may be appropriate before fitting a simple linear regression model.

We can perform transformations directly in the call to the regression function lm() as long as we use the 'as is' function I(). This is used to inhibit the interpretation of operators such as "+", "-", "*" and "^" as formula operators, and insure that they are used as arithmetical operators.

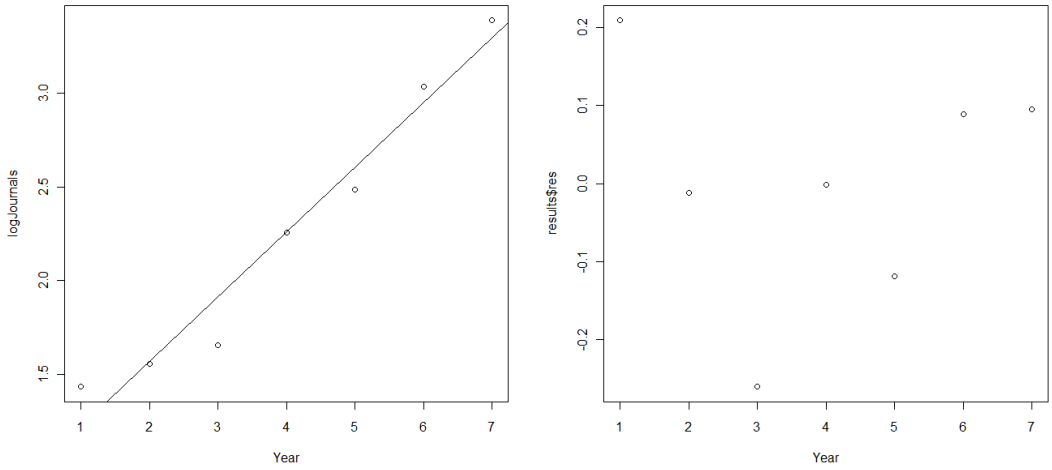
To fit a simple linear regression model using the logarithm of *Journals* as the response variable write:

```
> attach(Internet)
> results = lm(I(log10(Journals)) ~ Year)
> summary(results)

Call:
lm(formula = I(log10(Journals)) ~ Year)

Coefficients:
             Estimate   Std. Error  t value    Pr(>|t|)
(Intercept)  0.87690    0.14270     6.145      0.001659 **
Year         0.34555    0.03191     10.829     0.000117 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.1688 on 5 degrees of freedom
Multiple R-squared: 0.9591,     Adjusted R-squared: 0.9509
F-statistic: 117.3 on 1 and 5 DF,  p-value: 0.0001165
```

According to the output, the model is given by $\log(\hat{y}) = 0.8769 + 0.3455x$.

To make a scatter plot of year and log journals with the regression line overlayed and a plot of the residuals against year, type:

```
> plot(Year,I(log10(Journals)))
> abline(results)
> plot(Year,results$res)
```



The scatter plot of log(journals) and years appears linear and the residual plot shows no apparent pattern. It appears that a simple linear regression model is appropriate for the transformed data.

Next, suppose we want to use the model to predict the number of electronic journals at the end of 1998 ($x = 8$). To do so we need to first predict the log number of journals and thereafter transform the results into number of journals.

```
> coef(results)
(Intercept)        Year
  0.8769041   0.3455474
> log.yhat = 0.8769041 + 0.3455474*8
> log.yhat
[1] 3.641283
> yhat = 10^log.yhat
> yhat
[1] 4378.076
```

The predicted number of journals that will be available on-line by the end of 1998 is 4374.

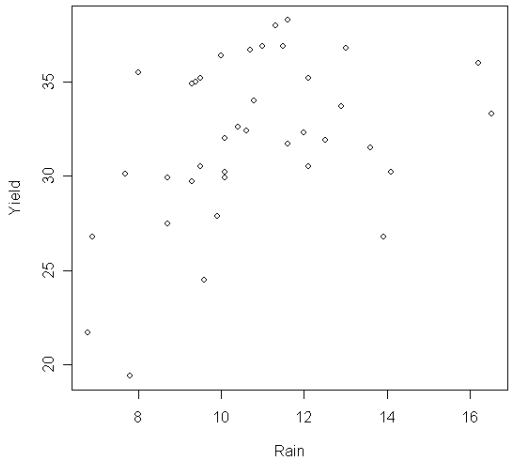
B. Polynomial Regression

Polynomial regression models are useful when there is reason to believe the relationship between two variables is curvilinear. In a polynomial regression model of order *p* the relationship between the explanatory and response variable is modeled as a p^{th} order polynomial function.

Ex. Data was collected on the amount of rainfall and the corn yield at a farm during the period 1890-1927. The data set consists of 37 observations on the three variables: Year, Yield and Rain.

To read in the data and make a scatter plot, type:

```
> dat = read.table("C:/W2024/Corn.txt",header=TRUE)
> plot(Rain,Yield)
```



There exists a clear curvilinear relationship between the variables which appears to be quadratic. Hence, we can fit a polynomial regression model of order 2.

```
> results = lm(Yield ~ Rain + I(Rain^2))
> summary(results)

Call:
lm(formula = Yield ~ Rain + I(Rain^2))

Coefficients:
             Estimate   Std. Error    t value   Pr(>|t|)
(Intercept) -5.01467    11.44158     -0.438     0.66387
Rain         6.00428     2.03895      2.945     0.00571 **
I(Rain^2)   -0.22936     0.08864     -2.588     0.01397 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 3.763 on 35 degrees of freedom
Multiple R-squared: 0.2967,     Adjusted R-squared: 0.2565
F-statistic: 7.382 on 2 and 35 DF,  p-value: 0.002115
```

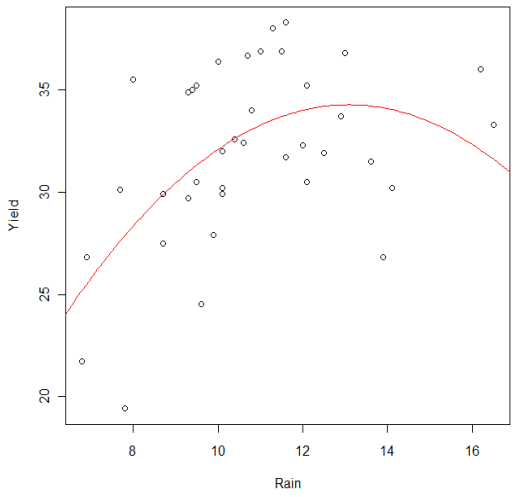
The fitted model is given by $\hat{y} = -5.01 + 6.00x - 0.23x^2$.

Next we make a scatter plot of year and yield overlaying the regression line.

```
> beta = coef(results)
> beta
(Intercept)        Rain           I(Rain^2)
 -5.0146670   6.0042835    -0.2293639
```

To plot the polynomial regression function use the following set of commands:

```
> plot(Rain,Yield)
%Define a vector (6.0, 6.1, 6.2, …..16.9 17.0) covering the range of the
        explanatory variable.
> vec = seq(6, 17, by=0.1)
> lines(vec, beta[1]+beta[2]*vec + beta[3]*vec^2, col='red')
```



The second order polynomial regression model appears to fit the data well.