Simple Linear Regression and Curvilinear Regression

Lab 4 R Notes: EXST 7014/15

Contents

0.1 Objectives

1. Use the **lm** function to fit a simple linear model
2. Check the assumption of homogeneous variances and normality test
3. Use the **lm** to fit curvilinear models.

Simple linear regression (SLR) is a common analysis procedure used to describe the significant relationship between two variables in such a manner that one variable can be predicted or explained by using information from the other. In the previous labs, we learnt how to evaluate the SLR model comprehensively by interpreting the ANOVA table, R^2, parameter estimates, residual plot, normality test and diagnostic statistics.

However, many systems encountered in research and practice exhibit curvilinear relationships instead of a simple linear relationship. Luckily, many curvilinear relationships can be expressed in linear relationships.

During the last lab, you might be aware that the homogeneity - that is assumes constant variability about the regression line - is a common violation of one of the assumptions of linear regression. If the variability increases as values of the predicted value increases then certain transformations are applied. Among the choices are the log, square root, and reciprocal transformations. Usually the need for one of these transformations is determined by examining the residual plot. If the residual plot is fan shaped then the heterogeneity of variance is assumed. Log transformation is the most commonly used to alleviate a problem with heterogeneity of variance. Using a log transformation implies that the underlying relationship is exponential. If the transformation works and the underlying relationship is exponential then the regression model should improve, and the residual plot should be more oval than fan-shaped.

0.2 Lab Setup

Run the following code to both install and load the required packages.

```
install.packages('olsrr')          # install the package that runs residual plots and check assumptions
library(olsrr)                     # Load the package
```

0.3 The Data

The dataset is from Chapter 8, Problem 10 in your textbook. We are trying to estimate the survival of liver transplant patients using information on the patients collected before the operation. The variables are:

- CLOT: a measure of the clotting potential of the patient's blood;
- PROG: a subjective index of the patient's prospective of recovery;
- ENZ: a measure of a protein present in the body;
- LIV: a measure relating to white blood cell count and the response;
- TIME: a measure of the survival time of the patient.

In this lab we will use the TIME as the dependent and ENZ as the independent variable. The data is available at http://statweb.lsu.edu/EXSTWeb/StatLab/DataSets/EXST7015/FW&M%20Data%202010/TEXT/DATATAB_8_31.TXT

```
#' The data link above is unavailable now
#' so download the data_lab4.txt file to your working directory
#' Create an object to host the data set
#'
#' @sep="" because the columns are seperated by 'space'
#'

patients <- read.table('data_lab4.txt', header = TRUE, sep = "")

str(patients)  # get a structure (description) of your dataset
```

```
## 'data.frame':    54 obs. of  6 variables:
##  $ obs : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ clot: num  3.7 8.7 6.7 6.7 3.2 5.2 3.6 5.8 5.7 6 ...
##  $ prog: int  51 45 51 26 64 54 28 38 46 85 ...
##  $ enz : int  41 23 43 68 65 56 99 72 63 28 ...
##  $ liv : num  1.55 2.52 1.86 2.1 0.74 2.71 1.3 1.42 1.91 2.98 ...
##  $ time: int  34 58 65 70 71 72 75 80 80 87 ...
```

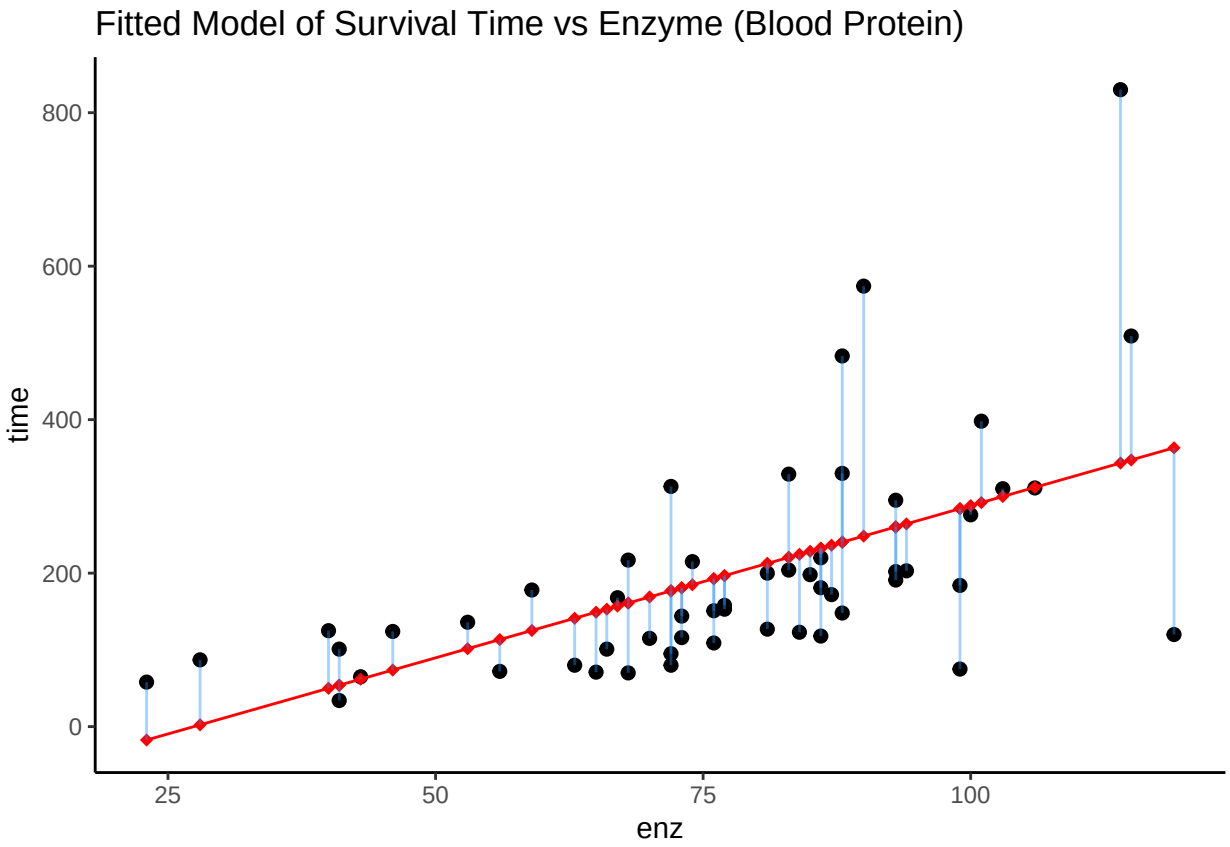
0.4 Fitting the Simple Linear Regression Model

$TIME = \beta_0 + \beta_1 ENZ + \epsilon$ where $TIME$ is the Y, ENZ is X, and ϵ is a random error term that is normally distributed with mean 0 and unknown variance σ^2. β_0 is the estimate of $Y - intercept$, and β_1 is the estimate of the slope coefficient.

```
#' Create an object called lm_patients (it can be any name)
#'      to host the model
#'
#'
#' Specify the model, time = B0 + enz (B1) using the lm function

lm_patients <- lm(time ~ enz, data = patients)
```

Visualizing fitted model with observations. The blue lines represent the errors for each fitted value. The red line is the fitted model.



0.4.1 Probing the lm function

The model created above contains a lot of information The object created to host the **lm** model can be subsetted (or extracted) with the following names:

```
names(lm_patients) # produces the call names of the lm function
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

Illustration of how to use the **names** of the lm function.

```
lm_patients$coefficients # extracts the coefficients (parameter estimates) of the model
```

```
## (Intercept)         enz
##  -108.71614     3.96678
```

Also certain sub-functions specific to the **lm** model can be applied to the model object (lm_patients)

```
 methods(class = class(lm_patients))[1:10] # extracts the 1st 10 functions
```

```
##  [1] "add1.lm"                    "alias.lm"
##  [3] "anova.lm"                   "case.names.lm"
##  [5] "coerce,oldClass,S3-method" "confint.lm"
##  [7] "cooks.distance.lm"          "deviance.lm"
##  [9] "dfbeta.lm"                  "dfbetas.lm"
```

An example is :

```r
confint(lm_patients) # produces 95% CI of parameter estimates
```

```
##                   2.5 %    97.5 %
## (Intercept) -232.564499 15.132220
## enz            2.417402  5.516158
```

Some global base R functions like **plot( )**, **summary( )**, **print( )** can be applied to the lm model.

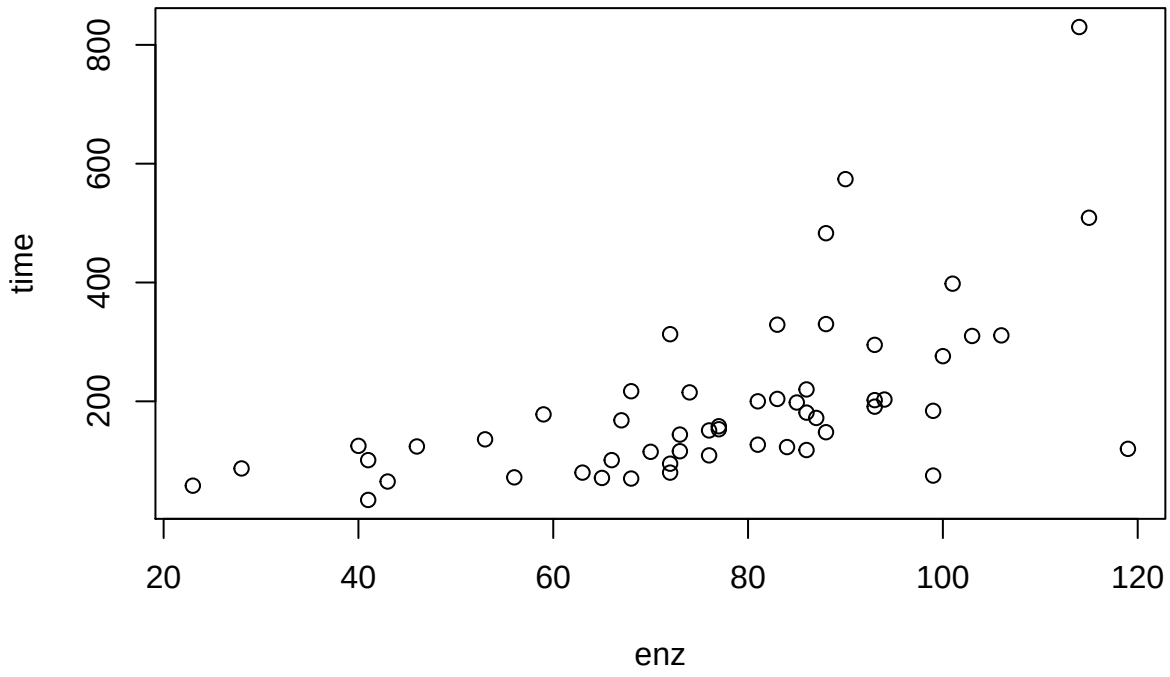
Example:

```r
print(lm_patients)
```

```
##
## Call:
## lm(formula = time ~ enz, data = patients)
##
## Coefficients:
## (Intercept)          enz
##    -108.716        3.967
```

```r
#' @plot does not have the data argument
#' so to avoid using the $ (indexing/subsetting) symbol
#'      the @with is used to attach the patients dataset
#'      for the @plot() function

with(patients, plot(enz, time)) # this produces a scatterplot of enz vs time
```



0.4.2 Assessing Homogeneity of Variance and Normality Assumptions

The residual plot can be used to detect various problems such as non-linear, non-homogeneous variances and outliers. If the data is of homogeneity of variance, most of residual points of the data randomly scatter around the mean residual (or zero line). If patterns like curvature (that is, non-homogeneity of variance) are detected in the residual plot, we may consider fitting a more complicated model.

Checking Homogeneity of Variance

```
#' The function below is from the olsrr package
#'
#' @ols_plot_resid_fit function plots the model residuals against
#'    the fitted values of the model
#' this function has one argument which is the name of the lm object

ols_plot_resid_fit(lm_patients)


#' Alternatively you can use the plot function from base R
#'
#' Applying the plot() on the lm object produces several diagnostic plots
#' the @which= can be used to extract a particular plot in this case,
#'       the plot for Fitted values against residuals
plot(lm_patients, which =1)
```

Checking Normality of the Residuals

This assumption is assessed by checking the normality of the **residuals**. Shapiro-Wilk is a popular statistics to evaluate normality of some data (in this case, residuals data). - Null Hypothesis: The residuals are normally-distributed - Decision Rule: Reject the null, IF the p-value is LESS THAN the significance level (say, 0.05) and conclude that the residuals are NOT not normally distributed.

```
#' @ols_test_normality also from the olsrr paxkage
#'
#' this function has one argument which is the name of the lm object

ols_test_normality(lm_patients)
```

```
#' Alternatively you can use the shaprio.wilk function from base R
#' Extract the model residuals @lm_patients$residuals

shapiro.test(lm_patients$residuals)
```

0.5 Fitting the Exponential Model

$log(TIME) = \beta_0 + \beta_1 ENZ + \epsilon$ where $TIME$ is the Y, ENZ is X, and ϵ is a random error term that is normally distributed with mean 0 and unknown variance σ^2. β_0 is the estimate of $Y - intercept$, and β_1 is the estimate of the slope coefficient.

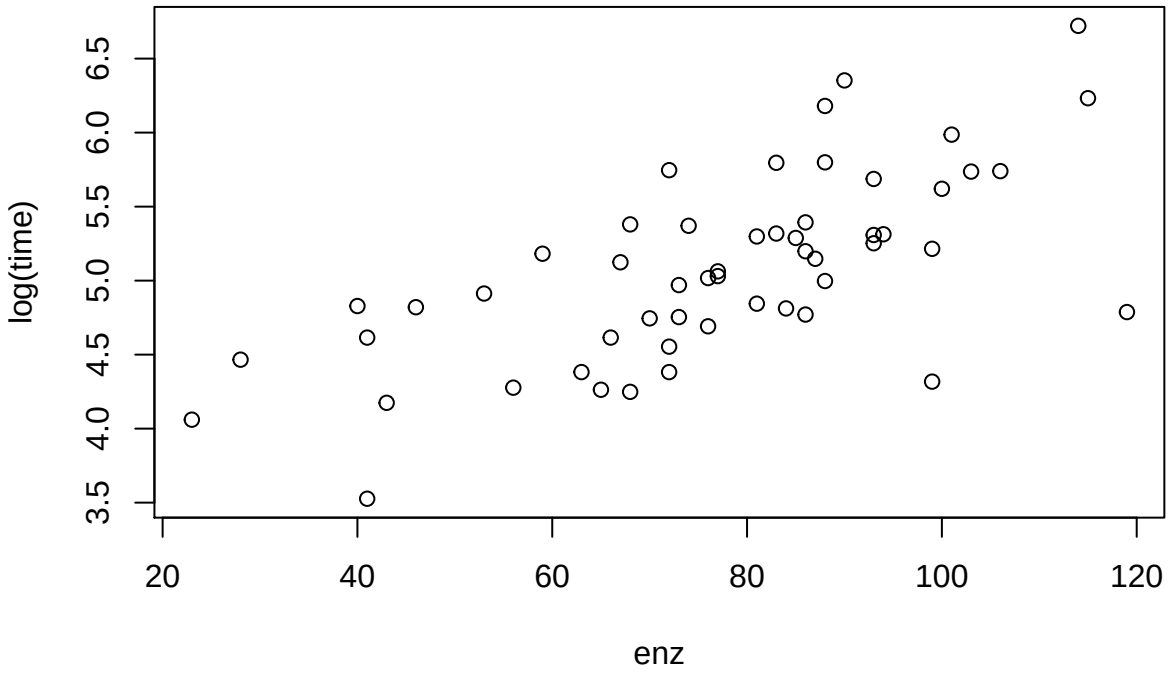
```
#' The model below fits the log(Y) against X
#' In R the natural log is the default for the function log()
#'
#' Specify the model, log(time) = B0 + enz (B1) using the lm function

log_patients <- lm(log(time) ~ enz, data = patients)
summary(log_patients)
```

```
##
```

```
## Call:
## lm(formula = log(time) ~ enz, data = patients)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.19415 -0.29725 -0.02198  0.34125  1.01853
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.558633   0.245526  14.494  < 2e-16 ***
## enz         0.019727   0.003072   6.423 4.12e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4753 on 52 degrees of freedom
## Multiple R-squared:  0.4424, Adjusted R-squared:  0.4316
## F-statistic: 41.25 on 1 and 52 DF,  p-value: 4.118e-08
```

```
with(patients, plot(enz, log(time)))
```



0.6 Lab Assignment

Your assignment is to answer the following questions by performing necessary analysis in either SAS or R. Only report or print necessary results.

0.6.1 Question 1

Make a scatter plot to show the relationship between TIME and ENZ. What is your observation? How about the scatter plot showing the relationship between Log-Time (that is the log transform of Time) and ENZ.

0.6.2 Question 2

Fit the simple linear regression model $TIME = \beta_0 + \beta_1 ENZ + \epsilon$. Write down the estimated regression function and examine the residual plot and normality test. Describe what you observed and make brief comments. **Hint:** you need to check the ANOVA table (that is the F-Statistic and its p-value on the last line of the **summary(yourmodel)** output)), parameter estimates tables, R-Square, residual plot and normality test.

0.6.3 Question 3

Fit the exponential model $logTIME = \beta_0 + \beta_1 ENZ + \epsilon$. Write down the estimated regression function. Does the model fit well? Why? **Hint:** you need to check the ANOVA table (that is the F-Statistic and its p-value on the last line of the **summary(yourmodel)** output)), parameter estimates tables, R-Square, residual plot and normality test.

Remember to attach your code

0.6.4 Question 4

Compare the simple linear model in Question 1 and the exponential model in Question 2, do you observe any improvements after conducting the exponential model relative to the linear model? Support your conclusion with details (such as R-Square, homogeneity of variance and normality test)