Multiple Linear Regression: Partial Correlations and Variable Diagnostics

Lab 6 R Notes: EXST 7014/15

Contents

0.1 Objectives

1. Use the **lm** function to fit multiple regression models
2. Get familiar with standardized regression coefficients and partial correlations
3. Perform some variable diagnostics and detect multicollinearity

In multiple regression, a number of variables can be involved and regressed on one another ($model \;\; Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon$).

The overall test of hypothesis of MLR is

$$H_0 = \beta_1 = \beta_2 = ... = \beta_p = 0$$

and the alternate is:

$$H_1 = at \;\; least \;\; one \;\; \beta \neq 0$$

Rejection of the null, H_0 implies that at least one of the regressors, $X_1, X_2, ..., X_p$, contributes significantly to the model.

In lab 5, we were introduced to the problem of multicollinearity which arises when there are either highly correlated predictors or when some **predictors** can be represented as a linear combination of the others. In this lab, an extreme case of multicollinearity will be discussed to further your understanding of the various diagnostics statistics (sequential parameter estimates, simple correlation, variance inflation factor (VIF) and condition index) introduced in the previous lab.

Since there are more than one independent variable in the MLR model, the question of the relative importance of the variables is of keen interest. We can assess the relative importance of the predictors by using partial SS F-test(Type II, III, IV) and t-test of regression coefficients. The larger the F-value or t-value (that is, the smaller the p-value), the more significant the corresponding variable (or predictor). In addition, **standardized regression coefficients and partial R^2 statistics** would be employed as tools to evaluate the relative importance of the individual variables.

Some of you might realize that the absolute value of regression coefficient is not a good predictor of relative importance of the variables. Why it happens is that, most often, the variables are not on the same scale or are of arbitrary scale, which leads to meaningless regression coefficient (Y units per X units). In such cases, the variables could be standardized with a mean=0 and variance=1. Then the standardized regression coefficients are obtained, which is the relative measurements of the importance of the variable.

In multiple linear regression, R^2 for overall model is the proportion of variation in dependent variable explained by all independent variables included in the model (SSModel/SSTotal). Likewise, a partial R^2 could be calculated for each individual variable, which measures the marginal contribution of one independent variable when all the other variables are already included in model.

0.2 Lab Setup

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

```
#' a function to only install needed but unavailable packages
#' and loads these packages after installation

ipak <- function(pkg){
    new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
    if (length(new.pkg))
        install.packages(new.pkg, dependencies = TRUE)
    sapply(pkg, require, character.only = TRUE)
}

# use function to install and load packages
packages <- c('olsrr', 'car', 'rsq')
ipak(packages)


#' olsrr   # package that assesses model fit and variable diagnostics
#' car     # package for Type II, III SS
#' rsq     # for Partial R-Square Type II
```

0.3 The Data

The dataset is from Chapter 6, Problem 18 in "Introduction to Regression Analysis" by Abraham and Ledolter @ 2006 Thomson Brook. This data set came from an experiment to investigate the amount of drug retained in the liver of a rat. Nineteen rats were weighted and dosed. The dose was approximately 40mg/kg of body weight. It can be expected that the liver is strongly correlated with body weight. After a fixed length of time the rat was sacrificed, the liver weighted and the percentage of dose in the liver was determined.

The variables are: bodyWT (body weight), liverWT (liver weight), DOSE and Y (Dose remained in liver). We will perform a multiple regression using Y as dependent variable and bodyWT, liverWT and DOSE as independent variables.

```r
#' Download the data_lab6.txt file to your working directory
#' Create an object to host the data set
#'
#' @sep="" because the columns are seperated by 'space'
#' @na.strings="." converts all missing values indicated by period . to NA

liver <- read.table('data_lab6.txt', header = TRUE, na.strings = ".")

str(liver)  # get a structure (description) of your dataset

#View(liver) # to view the liver dataset in RStudio's GUI pane
```

Just like in the dataset for the previous lab, there were some variables with missing values indicated by the **period** symbol[**.**] in this lab's dataset. In that lab we employed an approach that first converts such variables to characters and subsequently to numeric. That approach can be cumbersome if you have many variables, say 50, with missing values.

The argument **na.strings** = "." in the `read.table` function instructs `R` to convert to **NA** whenever it spots the **period** symbol[**.**] in a numeric variable. This approach is much better. Note that had the missing values been indicated by another symbol say, the **underscore** [_], then the argument would be **na.strings** = "_"

Notice also that the **dec** = "." was not specified because that is the default for the read.table function.

0.4 Analyzing the Data

For the purposes of this lab we are interested mainly in the following:

- Standardized regression coefficients
- correlation matrix of the parameter estimates ($\hat{\beta}_i$)
- partial R^2 type II
- collinearity diagnostics like the condition index
- variance inflation factor, VIF
- Sequential Parameter Estimates

0.4.1 Fitting the model

```r
#' fit the full model without preprocessing the variables - hence raw vars
fullRaw.lm <- lm(Y ~ bodyWT+liverWT+dose, data=liver)
summary(fullRaw.lm)
```

A few KEY things to note about the lm function:
- For each row with at least one missing value, it (the lm function) removes the ENTIRE row before it fits the model.
- All the predictors can be specified entirely with the period symbol, if the dataset contains precisely ONLY those predictors and dependent variable.
- The **lm** function returns a list containing several outputs, one of which is **model** which is a data frame containing the data used in fitting the model. Note this data (in our case, **fullRaw.lm$model**) MAY NOT be the same as the original data set (in our case, liver). Recall from the first bullet point that the lm function, by default, removes all rows with missing values before fitting the model.

0.4.2 Standardized Regression Coefficients

Over here the idea is to determine the relative importance of variables by the relative weight (that is, compare the absolute values) of their standardized coefficients. This can be done by 2 approaches:

Approach 1: Standardize all the variables (both dependent and independent) before fitting the model.

To do that you would need the **scale** function. **Caution:** By default, the **scale** function standardizes a vector (or variable) by considering all data points including missing values. This results in under-estimating the mean and over-estimating the variance of the variables when there are missing values.

Now, we also do not want to remove missing values in each column (or variable) individually before standardizing them. Doing so might result in different number of observations (or lengths) for each variable. So we need to , FIRST, manually rid our dataset of all missing values by removing the ENTIRE row whenever there is at least one missing value in the variables.

```
no_NAs <- complete.cases(liver)  # scans through your dataset and extracts an index of
                                 # the observations (rows) with no missing values
liver_noNA <- liver[no_NAs,]     # apply the index (row keys) to subset only observations
                                 # with non-missing column data
str(liver_noNA)

#liver_trans <- na.omit(liver)  # alternative approach
```

```
#' apply the @scale function to all columns in the @liver_noNA
#'   dataset by using the @lapply function
#'
#' the lapply function produces a list of the variables
#' convert list to data frame by using the @as.data.frame function
#' Store transformerd dataset as liver_trans

liver_trans <- as.data.frame(lapply(liver_noNA, scale))
str(liver_trans)
```

```
# Fit the model with standardized coefficients
fullTrans.lm <- lm(Y ~ bodyWT+liverWT+dose, data=liver_trans)
summary(fullTrans.lm)
```

Approach 2: Fit model and then standardize the regression coefficients

$$Standardized\ \ Coefficient = \hat{\beta}_i^* = \hat{\beta}_i \times \frac{\sigma_{X_i}}{\sigma_Y}$$

Recall from Lab 4 that we probed the **lm** function by exploring its various components

```
names(fullRaw.lm)
attributes(fullRaw.lm)
```

We can extract the coefficients of the model by applying the **coefficients** function to the model object.

```r
#' Assign an object called betas to store the model
#'      coefficients (or parameter estimates)

betas <- coefficients(fullRaw.lm)

#' Extract model variables
target <- 'Y'      # Specify Outcome Variable
predictors <- names(betas)[-1]   # Extracts the predictor names

#' Apply extracted model variable names to model data @fullRaw.lm$model
#' Use the @apply function to compute the standard deviations @sd
#'     for each column(indicated by the key 2)

sd_Y <- apply(fullRaw.lm$model[, target, drop=F], 2, sd)
sd_allXs <- apply(fullRaw.lm$model[, predictors], 2, sd)

#' Remove the intercept by using betas[-1]
Standardized.Betas <- betas[-1] * (sd_allXs/sd_Y)
Standardized.Betas
```

0.4.3 Correlation Matrix of Estimates

The correlation matrix of the parameter estimates (coefficients) can be easily extracted from the **summary** of the **lm** object by including the **corr=TRUE** argument.

```r
#' Produces the usual summary report of the lm object
#'    including the correlation matrix - which can
#'    be seen at the tail end of the summary report
summary(fullRaw.lm, corr=TRUE)


#' Subset the above output to report only the correlation
#'    matrix using the @$correlation

summary(fullRaw.lm, corr=TRUE)$correlation
```

Note: several other specific outputs can be extracted from the summary of the lm object. Run **str(summary((fullRaw.lm, corr=TRUE)))** to see the various other outputs.

Idea here is that variables with high correlation (say over ± 0.7) is an indication of multicollinearity.

0.4.4 Partial R^2 Type II

```r
#' the @Anova function below is from the car package
#'   and it is different from the base R @anova function
#'
#' The car Anova allows you to extract Type II, III SS
```

```r
#library(car)
Anova(fullRaw.lm, type=2)  # Type II SS for the Raw Model
```

The respective Type II SS for each predictor can be found under the column **Sum Sq**

```r
#' The @rsq.partial function is from the rsq pacakge
#'      it takes the model object as its argument
#'
#' it produces the partial R-Square type II
rsq.partial(fullRaw.lm)


#' Alternatively you can square the partial correlations from the
#'   @ols_correlations function to get the Partial R^2 Type II
(ols_correlations(fullRaw.lm)$Partial)^2
```

Usage details with some examples of the rsq package can be found at https://rdrr.io/cran/rsq/man/rsq.partial.html.

Key here is that, the bigger the Type II R^2 value for a particular variable the more 'important' it is.

```r
#' To get the semi-partial R-Square Type II Use
(ols_correlations(fullRaw.lm)$Part)^2
```

0.4.5 Condition Index & VIF

If condition index exceeds 30, multicollinearity might be a problem.
The value of VIF is expected to be 1 if the regressors are not correlated. If the value is very large, serious problems are suggested.

```r
#' all functions below (in this chunk) take the model object as its argument

vif(fullRaw.lm)  # Produces only VIF values
ols_eigen_cindex(fullRaw.lm) # Produces Condition Index without VIF values
ols_coll_diag(fullRaw.lm) # Produces both VIF and Condition Index
```

0.4.6 Sequential Parameter Estimates

That is sequentially add variables and refit the model. Order of addition of variables matter.

$$Model\ 1: \quad Y = intercept\ \ only\ \ model$$

$$Model\ 2: \quad Y = intercept + bodyWT$$

$$Model\ 3: \quad Y = intercept + bodyWT + liverWT$$

$$Model\ 4: \quad Y = intercept + bodyWT + liverWT + dose$$

- Recall from section 4.1 above that, the dataset used in fitting the model can be extracted using **fullRaw.lm$model**.
- Also we can specify the model formula as $Y \sim$. provided the data contains ONLY the variables intended to fit the model.
- We are going to use this simple idea to fit the sequential models above, and extract their coefficients using a simple **for-loop**.

```
results <- list() #generate empty list to host models

#' @length(fullRaw.lm$model) produces a number indicating the number of
#'    columns or models to fit
#'
#' @data = fullRaw.lm$model[1:i]) extracts only the relevant variables
#'      to fit the model at each iteration
#'
for (i in 1:length(fullRaw.lm$model)){
  results[[i]] <- coefficients(lm(Y~., data = fullRaw.lm$model[1:i]))
  # apply coefficients to extract coefficients of each model
}
results
```

0.5 Lab Assignment

Use PROC REG with appropriate options to fit the multiple linear model, $Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon$ and answer the following questions.

0.5.1 Question 1

Report the usual results of multiple linear regression (Hints: Hypothesis test results, Parameter estimates, regression function, and the assumptions for homogeneous and normality.)

0.5.2 Question 2

Is there any multicollinearilty? Why?

0.5.3 Question 3

Use RSTUDENT and Hat diag to check the outliers. And also use Cook's D, DFFITS, and DFFBetas to do influence diagnostics.

0.5.4 Question 4

In the output there are two columns called "95% CL mean" and "95% CL Predicted". Explain what their difference is.

0.5.5 Question 5

What is the partial R^2 type II for the variable DOSE. Can it be used to evaluate the importance of DOSE?

0.5.6 Question 6

Carefully exam the values of standardized regression coefficients and partial R^2 type II for individual independent variables, do you see the similar trends that you see in t-values in the t-test of regression coefficient? Make brief comments.