Computer lab 5 in MMS075, Feb 19, 2020

1. Re-do exercise 2 in exercise class 5a using a computer. That is, load the **Carseats** library and define a multiple linear regression model for predicting sales of car seats based on the variables **CompPrice**, **Income**, **Advertising**, **Price** and **ShelveLoc**, and plot the model. Once you have the Residuals vs Leverage plot on the screen, do the following steps to identify outliers and high leverage points:
    a. Draw a horizontal line, for example a blue dashed line, at values -3 and +3 for an easier identification of outliers;
    b. Compute the threshold of 2(p+1)/n recommended to identify high leverage points, and create a variable called **Threshold** containing this value. One way to specify the numerator and denominator in the threshold is to use the **$df** part of the summary, as follows: assuming that the model is named SeatModel, we have that **summary(SeatModel)$df[1]** equals p+1 and **summary(SeatModel)$df[1]+summary(SeatModel)$df[2]** equals n.
    c. Draw a vertical line at the value computed in part b as follows: **abline(v=Threshold,lty="dashed")**

    Based on this plot, what can you conclude about outliers and high leverage points?

2. Consider the model defined Exercise 14 on page 125 of ISL, with the following commands:
    set.seed(1)
    x1=runif(100)
    x2=0.5*x1+rnorm(100)/10
    y=2+2*x1+0.3*x2+rnorm(100)
    Model1=lm(y~x1+x2)

    a. During the exercise class, we computed the value inflation factor (VIF) value for this model based on the correlation. This time, compute the VIF values for Model1 with the **vif** function that is included in a library called **car**!

    b. Repeat the same procedure with using another number in the first command line instead of 1 (e.g. use **set.seed(87)**). Has the result changed? Re-do it again with set.seed(1) and confirm that you get back the original results.

    c. Use the **vif** function on the model defined in exercise 1 and check the results. Then remove the predictor ShelveLoc from the model and use **vif** again. Why is there a difference in the format of the outputs?

3. This exercise will do the optimization of the budget assignment in the advertising example using a computer. Download Advertising.csv from http://faculty.marshall.usc.edu/gareth-james/ISL/data.html and save it on the Desktop (i.e. in a computer-specific folder). Import the by using the menu in RStudio choosing File > Import Dataset > From Text (base)… Rename it as **AdvertisingData** while importing it to avoid a clash with the variable called Advertising in exercise 1. Attach the dataset for easier reference of its columns: **attach(AdvertisingData)**

We define a model with sales as response and TV, radio and their interaction as predictors:
AdModelInt=lm(sales~TV*radio)

Assume that there is a fixed total budget of $100 000 and we want to find the optimal way to divide this amount between TV and radio advertisements. This can be done by simply making predictions for all possible divisions, considering all 100 001 possibilities ranging from $0 to $100 000 on TV, as follows:

a. Define a variable called **TVBudget** containing a sequence of numbers from 0 to 100000 using either the **seq** or the **seq.int** function with appropriate arguments.

b. Since the variables '**TV**' and '**radio**' in the linear regression model are defined in $1000, divide each element of TVBudget by 1000 using the following command:
TVBudget=TVBudget/1000

c. Make a prediction of sales for each value of TVBudget using the pred function:
Predictions=predict(AdModelInt,data.frame(TV=TVBudget, radio=100-TVBudget))

d. Check where Predictions has its maximum value and what the corresponding prediction is:
which.max(Predictions)
max(Predictions)

Describe the optimal division of the total budget and the maximum estimate for sales in words and in terms of $, respectively sold units instead of variable values!

Now repeat and appropriately modify the above steps to find the optimal budget distribution if the available total budget is $50 000 and also for a total budget of $200 000.

4. Consider the logistic regression model predicting the probability of credit card default based on balance, as discussed in the lecture. The data that this model is based on can be found in dataset '**Default**' the **ISLR** library, and you may want to attach the dataset before proceeding to the further steps. Explore the dependence of default probability on balance via the following tasks:

a. Understand the **Default** dataset better by checking its summary. How many people are there in the dataset in total? How many have been defaulted? What is the percentage of students in the data?

b. Define the model in R using the following command:
glm(default~balance,family="binomial",data=Default)
In this command, glm stands for generalized linear model and family=binomial is needed to specify that we want to use logistic regression.

c. Display a summary of the model to see the coefficient estimates.

d. Make a prediction for the following balance values: $1000, $1500, $2000, $2500. This can be done by using the **predict** function, including the usual arguments (i.e. model name, data frame to perform the prediction for) plus an additional argument

writing **type="response"**. For your own understanding, check what you get as result if you do not include the **type="response"** argument!

e. Plot a function with balance on the x-axis and the predicted probability of default on the y-axis. This can be done using the curve command, as follows:
curve(predict(BalanceModel,data.frame(balance=x),type="response"), from = 500, to = 3000,xlab="Balance ($)",ylab="Estimated probability of default",cex.lab=1.5,cex.axis=1.2)

In this command, the first argument of curve is an expression that depends on x, and the from and to arguments specify the range of x-values that will be considered in the construction of the curve. The arguments **xlab** and **ylab** specify the axis labels and **cex.lab** sets the size of these labels. Finally, **cex.axis** sets font size for the numbers on the x- and the y-axis.

5. Let us now try predicting the probability of credit card default based on student status. Use the **glm** command as in exercise 5 to fit a model and display its summary.
    a. What are the coefficients of the model? Does being a student increase or decrease the probability of being defaulted?
    b. Compute the estimated probability of defaulting for students and also for non-students!
    c. Can you plot a curve similar to the one prepared in part e of exercise 5?

6. Finally, include all variables in the model and answer the following questions:
    a. Which variables are significant in this model and which are not?
    b. What are the coefficients of the model? Does being a student increase or decrease the probability of being defaulted? Is there any difference compared to part 5a?
    c. Compute the estimated probability of defaulting for a non-student with credit card balance of $1000 and annual income of $30 000!
    d. Estimate the probability of defaulting for a student with the same credit card balance and annual income values as in part c!

7. If you would like to ask something or give feedback, feel free to talk to me or enter your question/feedback at www.menti.com, using the code 71 36 17.