Lab 10: Multiple linear regression, dplyr, ggplot2

The following worksheet is due by 8 pm Wednesday March 29th. You can find the submission dropbox in Brightspace by clicking on Content − > Lab Content.

0. Open a new R Markdown file.

Note: Your worksheet is to be submitted as an R Markdown file (you can knit it to HTML and then convert it to PDF, or you can knit it to PDF if you have LaTeX on your computer, or you can knit it to Word and then convert that to a PDF).

1. Install the following packages: "dplyr", "ggplot2", "corrplot", "car", "lmtest", and "caret". Then load the dataset "stock-Canada.csv". You can find a description of the dataset in the file "stock-dictionary.csv". Your task is to predict the value of the stock. Please explain the output, your solutions, and the given codes.

(a) Clean the data by removing null and missing values. You may also choose to remove columns that have a high number of missing values.
(b) Convert the "REF_DATE" column to a date format using the "mutate()" function as follows:

mutate(REF_DATE = as.Date(paste0(REF_DATE, "-01"), format = "%Y-%m-%d"))

Note: This code uses the dplyr package's "mutate()" function to create a new column called "REF_DATE" in a dataframe, where "REF_DATE" is an existing column in the same dataframe. The purpose of this code is to convert the "REF_DATE" column, which is currently in a character format, into a date format.

The "as.Date()" function is used to convert the character strings in the "REF_DATE" column into date format. The "paste0()" function is used to concatenate the year-month values in the "REF_DATE" column with "-01", which represents the first day of the month, creating a new string in the format "YYYY-MM-01". This new string is then passed to the "as.Date()" function as the first argument, which converts it to a date format.

The "format" argument in the "as.Date()" function is used to specify the format of the input string. In this case, the format is "%Y-%m-%d", which indicates that the input string is in the format "YYYY-MM-DD". Since the input string only contains the year and month, "-01" is added to represent the first day of the month.

(c) Process the data using the "summary()" function and calculate the standard deviation for the column that has numeric values.
(d) Use the "ggplot()" function to visualize the relationship between each column and the value of the stock.
(e) Split the data into 80% train and 20% test using the following code:

```
# Split the data into training and test sets
set.seed(123)
train_index <- sample(1:nrow(dairy_stocks), size = round(0.8 * nrow(dairy_stocks)))
train_data <- dairy_stocks[train_index,]
test_data <- dairy_stocks[-train_index,]
```

Note: To split the data, you can use the sample() function and specify the percentage of data in each split. For example, "size = round(0.8 * nrow(dairy_stocks))" means 80% of the data will be used for training, and the rest will be used for testing.

(f) Build a multiple linear regression model to predict the value of the stock. The data type of some columns is not numerical (they are categorical). It is acceptable to consider those columns in the model, but in a data science project, you need to consider the data type.

(g) Identify variables that should be removed from the model.