Task 2

Write an R function RegrMean that calculates regression coefficients with repeated measures using lm function from R-base. Response is for the regression should be a mean of repeated measures.

Feel free using any number of functions to construct the solution. Feel free to create any classes you need. Try minimizing use of extra libraries. Do not use global variables. Please comment your code as much as possible.

Function inputs:

df – a data frame with the data. Each column is either a factor or response. Each record is an observation. Some factors are continuous (numeric data type) some factors are categorical (factor data type).

Example:

x1	x2	x3	x4	xKEYWORD6	KEYWORD_A
500	4.1	red	5.546	13564	56
600	1.3	blue	5.546	654	0.56
700	4.5	red	5.5	456	0.54
600	1.3	red	5.58	1567	1.57
500	4.1	blue	5.5	1.555	1200.2

Here each raw is an observation. Columns x1, x2, x3, x4 are regression factors. Columns xKEYWORD6 and KEYWORD_A are repeated measures. In this task, the response Y for the RegrMean function regression is Y = mean(xKEYWORD6, KEYWORD_A). If df has just one column with KEYWORD in the column name, e.g. KEYWORD_A, then the response Y is that column (Y = KEYWORD_A)

KEYWORD – a string. Data frame df columns with KEYWORD in the name are response measures. Data frame df columns without KEYWORD in the name are factors.

Example:

The data frame df columns names x1, x2, x3, x4 do not include KEYWORD. These columns should be treated as regression factors. The data frame df columns xKEYWORD6 and KEYWORD_A have KEYWORD in its name. These columns should be treated as simultaneously measured responses.

regressionOrder – an integer (1 or 2). The order of regression.

Example:

If regressionOrder = 1 then

```
Y = intercept + coef1 * x1 + coef2 * x2 + coef3 * x3
```

If regressionOrder = 2 then

```
Y = intercept + coef1 * x1 + coef2 * x2 + coef3 * x3 + coef4 * x1 ^ 2 +
coef5 * x2 ^ 2 + coef6 * x3 ^ 2
```

twowayInteractions – a Boolean variable (TRUE or FALSE). Indicates if first order two-way interactions should be included into regression equation.

Example:

If regressionOrder = 1 AND twowayInteractions = TRUE then

```
Y = intercept + coef1 * x1 + coef2 * x2 + coef3 * x3 + coef4 * x1 * x2 +
coef5 * x1 * x3 + coef6 * x2 * x3
```

If regressionOrder = 2 AND twowayInteractions = TRUE then

```
Y = intercept + coef1 * x1 + coef2 * x2 + coef3 * x3 + coef4 * x1 * x2 +
coef5 * x1 * x3 + coef6 * x2 * x3 + coef7 * x1 ^ 2 + coef8 * x2 ^ 2 +
coef9 * x3 ^ 2
```

Function outputs:

dfRegrMean – a data frame with regression results. The data frame should have the following columns:

Term – the column with the regression equation terms (“intercept”, “x1”, “x2”…, “x1 * x2”, “x1 * x3”,… “x1 ^ 2”, “x2 ^ 2”,…). The term names are specific to the data frame df. The first order term names are the same as the factor columns. The interaction terms are the factor names with the Asterix sign in between (like “x1 * x2”). The second order terms are the factor names with “^ 2” sign added (like x1 ^ 2)

Coef – the column with the regression coefficients for the respective terms. The respective coefficients should be in the same row as the term names.

SEcoef - the column with the standard error of the coefficients (estimates the variability between coefficient estimates that you would obtain if you took samples from the same population again and again.)

tvalue - the column with t-values of the coefficients. tvalue = Coeff / SEcoef

pvalue – the column with p-values for each regression coefficient.

fvalue – the column with F-values for each regression coefficient.

VIF - Variance Inflation Factor for each term `dfRegrResiduals` – a data frame that is equal to the source data frame `df` plus (*i.e. cbind()*) the following column

`Residuals` – the column with regression fit residuals (the difference between an observed value and the corresponding fitted value)

`StdResiduals` – the column with standardized residuals (value of a residual divided by an estimate of its standard deviation.)

`Rsq` – R-squared (is the percentage of variation in the response that is explained by the model.)

`Rsqadj` - R-squared adjusted (the percentage of the variation in the response that is explained by the model, adjusted for the number of predictors in the model relative to the number of observations.). Adjusted R-squared is calculated as 1 minus the ratio of the mean square error (MSE) to the mean square total (MS Total).

`MSE` – the mean square error

`SSE` – the sum of squares of the residual error

`flag` – a Boolean variable. `flag` = TRUE if the analysis was completed without errors. `flag` = FALSE if the analysis was not completed for any reason

`errorMessage` – (optional!) a string with an explanation why `flag` = FALSE. If `flag` = TRUE then `errorMessage` an empty sting.

Hints (feel free to disregard):

Hint 1. You may like to construct a string with the regression equation using `paste()`. The convert this string to a formula using `as.formula()` and pass it to the R regression function.

Hint 2. You may like use (x1 + x2 + x3) ^ 2 constructions to create interaction terms x1 * x2, x1 * x3, x2 * x3

Hint 3. Most of input df data frames will have 10 factors or less, 5 responses or less, 1000 observations or less. You may like to keep in mind this information constructing the solution.

Typical `df` will look like

x1	x2	x3	x4	xKEYWORD6	KEYWORD_A
100	1.3	red	5.546	13564	52
200	0	blue	5.546	654	0.54
300	1.3	red	5.5	4156	0.5
400	0	red	5.546	1364	57

500	1.3	blue	5.546	6524	0.56
600	0	red	5.5	4562	0.4
700	1.3	red	5.546	136	55
800	0	blue	5.546	6542	0.54
900	1.3	red	5.5	4563	0.52