Exploring the Performance of Linear Regression in Key Driver Analysis for Likert-Response Variables: A Simulation Study

In partial fulfillment of the requirements for Statistics 196.1 – Advanced Statistical Computing

Benabaye, Isabella M.
Cabral, Anne Louise D.
Dayta, Dominic B.
Donato, Patricia Rose B.

Faculty:
Michael Van Supranes

17th December, 2016

Abstract

A primary concern regarding the use of Key Driver Analysis (KDA) in market research is its lack of justification in statistical literature. The most popular technique, using multiple linear regression (MLR), is questionable for it violates theoretical assumptions in the nature of the data used. The present paper uses simulations to test MLR and Shapley regression, a method that that has been cited as the more robust although computationally-intensive, for finding key drivers. Situations considered involved the presence of multicollinearity and the omission of an important variable. Shapley performed better than MLR in most cases when the main drivers are easily discernible in importance but even then Shapley performances were not a long way above that of MLR, deeming the former's complicated process impractical. In the cases where the importances were to some degree indiscernible, neither MLR nor Shapley were able to find the correct key drivers.

Keywords

Key Driver Analysis, Multiple Linear Regression, Shapley Regression, Market Research

Introduction

Motivation

A primary function of market research consists of the improvement of products and services. However, before making any improvements, it is essential to know the importance of a product's attributes. Perhaps the company would like to know which features of its services affect customer loyalty and satisfaction the most (e.g. affordability, employee courtesy). Answering this means uncovering *key drivers* of the customers' brand attitude, or in another instance, their likelihood to recommend the products, etc. From here, among the list of the product's attributes, the company would then know which to prioritize the next time it makes improvements on them.

One popular method to identify such drivers is called the Importance-Performance Analysis or the Key Driver Analysis (KDA). Two popular techniques (among many others) to compute KDA estimates include: multiple linear regression (MLR) and Shapley regression. Between the two, MLR is more commonly used for its relative ease and wide availability of documentation, while Shapley, being the most recently developed method, is computing-intensive but robust especially in the presence of multicollinearity ("Driver (Importance) Analysis").

Despite the methods being widely used in the industry, there is still no clear statistical basis on whether these methods are valid or not, which is a concern since measurements taken for the variables or questions are ratings and therefore not continuous. Also lacking in documentation is whether a specific method is better than the other. This simulation study is interested in which scenarios MLR works better than the Shapley regression, and whether MLR's estimates are close enough to Shapley's (being robust), making it more preferable than the computing-intensive Shapley regression.

Significance of the Study

Through KDA, companies can be more informed regarding, for example, their customers' satisfaction, and consequently on which important factors they ought to focus their resources. Moreover, it can also be used for making policies that will improve or maximize the performance of a product or a service, or for new customer acquisition strategies. However, performing KDA is not always achievable, especially for companies with limited resources. This is particularly the reason why Shapley technique is almost always avoided despite being considered as the "state of the art".

In particular, MLR and Shapley regression have their own strengths and weaknesses. For instance, the linear regression technique is relatively easy to execute, however, it may cause unusual results when there is strong correlation among the predictor attributes. On the other hand, the Shapley regression, although it is a computing-intensive method, provides an estimator for measuring the importance of attributes which is robust to multicollinearity. Through this study, it might then be possible to show that using linear regression may still be preferred than the Shapley under certain scenarios. It may also be possible that results of linear regression be observed as being nearly close to or even better than that of Shapley.

With the study's results, it aims to clarify whether MLR is better than Shapley or vice versa, in terms of accuracy and most especially of practicality. Hence, users of such techniques will have an idea of what to expect and will be more confident of the results.

Problem and Objectives

In this study, the researchers intend to find out how well the linear regression can determine or predict the key driver(s). Due to its being computationally intensive, the Shapley regression is not as widely used as MLR, however it is often described as being "state of the art." The researchers wish to put this claim to test by observing and comparing the performances of the MLR and Shapley under certain scenarios (e.g. multicollinear regressors). For each of these scenarios, this study aims to determine the method that best and most efficiently provides the key driver(s) of the business outcome of interest ("Driver (Importance) Analysis").

Scope and Limitations

The simulation study will consider only four distinct scenarios based on combinations of the presence/absence of two modelling issues: omitted questions and multicollinearity. For scenarios with multicollinearity, only a high level will be considered. Varying levels of multicollinearity will not be explored. Also, in order to remain economical on computing time and resources, only 400 simulations will be done, each using a simulated data set with a hundred observations.

With regards to the simulation itself, certain limitations were imposed by the algorithm used. For one, the Mean Mapping Method algorithm used in the present paper requires a pre-specified marginal probability mass function for the k levels or outcomes of the p random variables. It would be reasonable to suppose that the need for a predetermined PMF has in some way crippled the flexibility of the simulation study. Nevertheless, considering the tedious nature of other simulation techniques that do not make use of any distributional assumption (such as presented in Biswas, 2004), the researchers decided to retain this limitation, simply resolving to use various distributions among the variables simulated in order to closely approach real world data.

One more limitation the prespecified PMFs imposed on the study was lowering the number of responses of the likert response variables from the best 7 (based on simulation studies by Jamal, 2014) to 5, to make manually distributing the PMFs easier upon calibration. The Mean Mapping Method also requires the use of a positive semidefinite correlation matrix. This imposed another limitation on the study. The researchers had to find a symmetric combination of hypothetical pairwise correlations that would together yield eigenvalues greater than or equal to zero. But in Wicklin (2013), not all positive semidefinite matrices would lead to a convergence in the objective equation presented in Kaiser et al (2011) in producing the multivariate normal simulation from which the ordinal values will be generated. The correlation matrix used in the study was the first that the researchers were able to generate that lead to a convergence, and thus no other correlation matrices aside from the identity matrix (for the case of no collinearity) were explored due to the difficulty of generating just one.

Review of Related Literature

This paper focuses on the issue of wide usage of linear regression methods in finding key drivers in survey data, even when the responses happen to be ordinal in nature (e.g. in a Likert scale). Osborne and Waters (2002) cite normality as an important assumption for the variables to be included in linear regression analysis. Failure to satisfy this assumption have been found to distort relationships and significance tests. Normally distributed data are continuous in nature and thus a problem is expected to arise when linear regression is performed on discrete variables. On a strictly theoretical perspective, using linear regression for survey data whose responses are typically in Likert scales, should not be permissible.

However, Winship and Mare (1984) cite multiple studies that argue using ordinal variables as if they were continuous may be permissible for under flexible and robust methods that the biases entailed may be safely ignored. The same paper cites as well arguments to the contrary, while themselves showing that the linear regression tends to give distorted results. Thus this paper will explore the use of linear regression in the context of performing KDA, the extent at and scenarios in which it can identify key drivers among regressor variables influencing regressor variables, all of which are assumed to be coming from a Likert scale.

An important aspect to explore in later formulating the methodology of this paper is finding a proper calibration for a Likert scale survey. Likert (1932, cited in Allen et al, 2007) himself suggests the use of wide scales. Commonly used scales in most research papers are five and seven choices in length. Opinions vary whether even-numbered scales are better in order to prevent neutral responses, or odd-numbered scales are optimal for the very fact that they allow for undecided answers. However, Jamal (2014) has shown that seven-point scales are most optimal, in the sense that they result in lower measurement error and higher precision compared to a five-level scale.

A common pitfall in the use of linear regression in KDA is when the predictor variables exhibit some level of multicollinearity. It is only intuitive to expect that certain aspects of a service or establishment may actually be influencing one another. Linear regression assigns effects to multicollinear variables unevenly, not to mention that in the presence of perfect multicollinearity the design matrix $\mathbf{X}^T\mathbf{X}$ is singular, making the estimated regression model done under ordinary least squares indeterminate. Even if the design matrix is invertible, any computed inverse will be ill-conditioned, and the resulting beta coefficients unstable.

Demonstrations have shown (Hansa Marketing Services, 2014) that a problem with multicollinearity may be assessed through improved regression methods for KDA that take into account proper decomposition of R^2 in models of all-possible-combination of the explanatory variables (e.g. Shapley regression). Moreover, the same demonstration has shown that ordinary regression analysis has a tendency to underestimate the importance of key drivers.

Methodology

Data Simulation

A hypothetical survey will be simulated, consisting of five questions, four of which will be used to explain (regressors) the responses to the fifth question (response variable). An example of such a survey in real-world settings might be asking for a customer's rating on four main aspects of a service, and afterwards asking how likely they are to avail the service another time. The study assumes all questions on the questionnaire will have responses ordinal in nature, particularly following a uniform Likert scale of five responses. The choice of this scale despite results showing the seven-point scale as more optimal is due to the requirement of a predefined probability mass function for each response in each variable for executing the data simulation algorithm. A shorter scale will be easier to distribute, but too short a scale might fail to mirror real life surveys.

The performance of ordinary regression analysis in finding the importance of key drivers in the simulated survey will be observed under various scenarios, such as in the presence of multicollinear regressor variables. Shapley regression will be done simultaneously for comparison.

To thoroughly examine the performance of the two methods, the researchers considered three overall cases, wherein 1) the key driver is evident among the questions, 2) many questions are strong drivers, but key driver is still fairly evident, and 3) all questions are strong drivers,

but the most influential is still considered as the key driver. For each of the two cases, the following scenarios will be simulated:

Table 1: Cases considered in the simulation study

Nature of Data	Nature of Key Drivers	Multicollinear	1 Question Omitted[1]
Likert-Type, 5-point scale	Strongly Evident	Yes	Yes
		Yes	No
		No	Yes
		No	No
	Fairly Evident	Yes	Yes
		Yes	No
		No	Yes
		No	No
	Indiscernible	Yes	Yes
		Yes	No
		No	Yes
		No	No

[1] *Omitted question is relevant, but it is NOT the key driver itself.*

In each scenario, 400 datasets will be simulated, each with a set of 100 responses. For the case of no omitted questions, four regressor variables will be simulated, with one response variable, all of which will be incorporated into the model. For the case of omitted questions, one of the four regressor variables will be omitted, with only the remaining three incorporated into the model. This is in order to generate the case of a variable with a significant contribution in explaining the variation in the response variable, but which has not been identified and included by the researchers.

Generating Multicollinear Ordinal Regressor Variables

Generating hypothetical surveys for this study requires simulating ordinal $p > 2$ variables $X_1, X_2,...,X_p$ each representing responses to questions coded on a Likert scale of values 1 to 5. For the case of no omitted questions, $p = 4$ response variables will be simulated. On the other hand, for the case of omitted questions, $p = 3$ response variables will be used, with the third as the "omitted" question. The p simulated variates must also follow a specified correlation structure, for the case of generating multicollinear variates.

Algorithms in simulating correlated K-level ordinal variables are abundant in literature. In Demirtas (2006), a possible method for doing this requires first dividing each ordinal variable into a binary variate X^{BIN} with values 0 or 1. Such that an ordinal variable X with levels 1,2,..K is recoded into X^{BIN} variable whose value is 0 when X = 1,2,..., $K/2$ and 1 when X = (K/2)+1,...,K. Binary data are simulated and then recoded into their K-level ordinal counterparts using proportions. Biswas (2004) lists down multiple possible alternative algorithms for use in this scenario. However, the algorithms presented would be arduous to code in SAS, and would work better in a more object-oriented environment such as R. Kaiser, Träger and Leisch (2011) provide an algorithm called the mean-mapping method that has been shown to outperform the algorithm in terms of being flexible to more initial correlation structures. The mean mapping method is used in this paper, using code adapted from Wicklin (2013).

The mean mapping method begins with grouping the p desired ordinal variables into pairs (X_i, X_j) and finding the root to a complicated equation that eventually yields the

cumulative distribution function of the bivariate normal distribution for their continuous counterparts (Y_i, Y_j). From this we solve the pairwise correlation for the two continuous variables. If we let **C** as the matrix of pairwise correlations c_{ij}, the *p* continuous variables are simulated from the multivariate normal distribution as $(Y_1,Y_2,..,Y_K)$~MVN(**0**, **C**).

The method requires prespecification of a probability mass function (PMF) for each value of the *p* ordinal variates. An advantage of this is that the *p* responses may be simulated in a way that the distribution of their responses will not all be alike (as might be expected in real data). Wicklin (2013) provides a helpful illustration for coding the simulation on SAS. Given the PMF of an ordinal variable $p_X(X=x)$, the algorithm will find the cumulative distribution F(X) and find standard normal quantiles such that for a value *c* of X: $\Phi(c) = F(X < c)$ where Φ is the cumulative standard normal distribution. It is clear that each of the (K-1) values of X will have a corresponding standard normal quantile (the Kth will have a cumulative probability of 1 and therefore no corresponding quantile in the standard normal distribution). The standard normal distribution will then be divided into K regions I_1, I_2,...,I_K delimited by these quantiles.

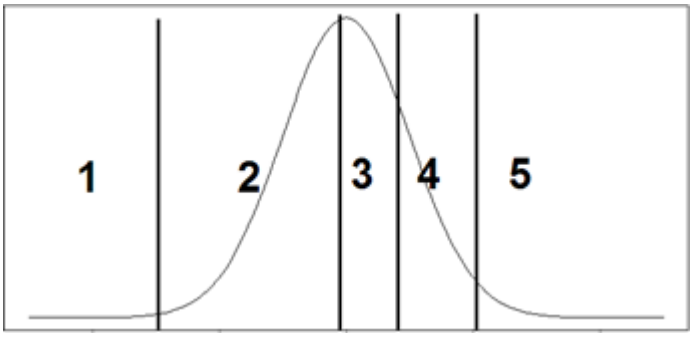

Figure 1: An illustration for dividing N(0,1) into quantiles

The simulated continuous variables will be converted into their ordinal counterparts by comparing each value and finding the region in which they fall based on the standard normal quantiles. The resulting ordinal variables will have a correlation structure approaching the desired values. How far the resulting ordinal correlation structures are from their original values after the mean mapping algorithm will be explored in the discussion of results.

As a restriction, the study will not explore varying levels of multicollinearity. For the general case wherein multicollinearity is present, a high level of collinearity will be immediately considered. However, for the special case in which an omitted key driver is highly collinear with a present weak driver, in order to highlight this association all other correlations will be diminished to moderate levels. Similarly, for collinear key and weak drivers that are both present (another special case), all other correlations will be reduced to moderate levels.

In assessing the ability of ordinary least squares regression in finding the key driver, we formulate a hypothetical regression model for the fifth question Y. However, Y must also be in the form of a Likert scale response on five levels. To do this, we form a hypothetical regression model in which the coefficients sum to 1, without inter cept, and add a random standard normal variable as random error. The hypothetical models

$Y_i = FLOOR(0.05\ X1_i + 0.15\ X2_i + 0.20\ X3_i + 0.60\ X4_i + \varepsilon_i)$ Case 1: Main Drivers Strongly Evident
$Y_i = FLOOR(0.10\ X1_i + 0.25\ X2_i + 0.30\ X3_i + 0.35\ X4_i + \varepsilon_i)$ Case 2: Main Drivers Fairly Evident
$Y_i = FLOOR(0.20\ X1_i + 0.23\ X2_i + 0.27\ X3_i + 0.30\ X4_i + \varepsilon_i)$ Case 3: Main Drivers Indiscernible

where FLOOR(.) is the greatest integer function and $\varepsilon_i \sim N(0,1)$.

The function inside the FLOOR(.) function will generally have values around 1 to 5, except for cases wherein the contribution of the error term is large. To keep the values around the 5-point Likert scale, a limit will be imposed such that

$$Y = 5 \quad \text{if FLOOR}[f(\underline{X})] > 5$$
$$Y = 1 \quad \text{if FLOOR}[f(\underline{X})] < 1$$

The result of the processes above will be a data set of five variables, each coded on the five-point Likert scale.

Key Driver Analysis Using MLR

For each data set, MLR will be performed using the generated independent variables. In Sambandam (2001), one output of interest from MLR is the relative importance of the independent variables expressed in the form of coefficients or beta weights. Beta weights are used to identify the variables that have the most impact on the dependent variable. The *importance* of the i^{th} variable or question which is the proportion of the absolute values of the estimated coefficients will be computed, i.e., $P_i = |\beta_i| / \sum_i |\beta_i|$. Then, the largest of these calculated proportions will determine the key driver for that particular data set. After this process is done for the 400 datasets, the variable which has the most number of maximum *importance* (i.e. most frequent key driver) will be the key driver that MLR considers under that specific scenario. The variable with the second most number of maximum *importance* will be the second driver. The frequency of each question being identified as the key driver and the second driver will be taken. The estimated coefficients from each of the regression runs for the 400 datasets will also be outputted to be analyzed later.

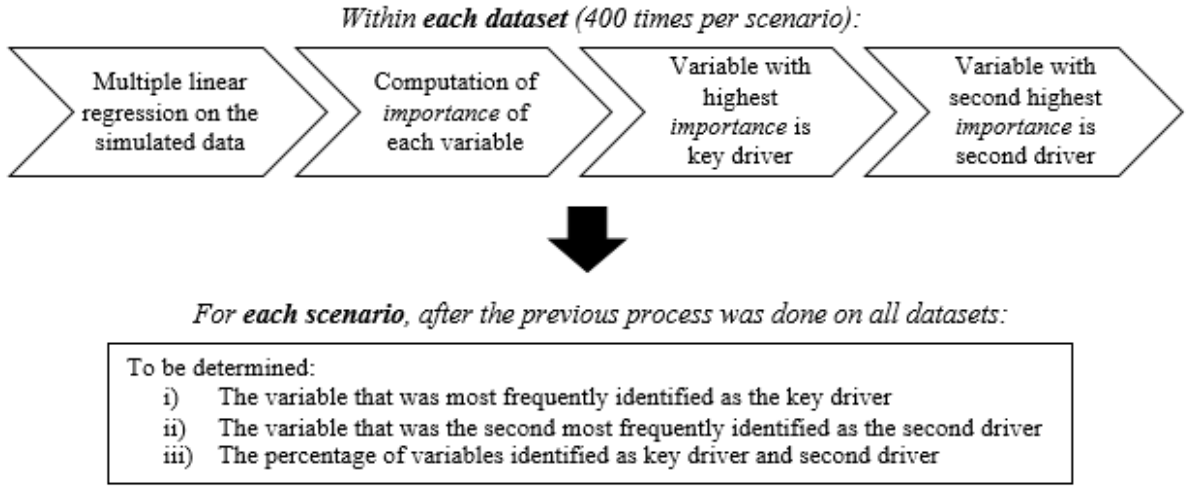

Figure 2: A flowchart of the steps to be followed in KDA through MLR

Key Driver Analysis Using Shapley Regression

Shapley regression, unlike the regular MLR approach, considers the average marginal R^2 for each regressor, in which the regressor having the largest marginal R^2 is considered as the key driver. Though tedious, this method addresses the consequences of multicollinearity among the regressors. For this case, there are four independent variables whose individual contributions, called Shapley values, the researchers will be computing for.

First, all combinations of regressors (individual, pairwise, and so on) will be run against the dependent variable. R^2 values are then extracted from each regression. This is done for the 400 datasets for all the 4 scenarios.

For four regressors, the table below shows all the possible combinations for regression, amounting to 15 regression equations for each dataset. Consequently, there are 15 R^2 to be compiled (horizontally) for each dataset.

Once all R^2 are compiled, the Shapley values for the independent variables A, B, C, D are computed as follows:

$$Shapley\ Value\ (A) = \frac{\left\{(R^2_{ABCD} - R^2_{BCD}) + \left[\frac{(R^2_{ABC} - R^2_{BC}) + (R^2_{ABD} - R^2_{BD}) + (R^2_{ACD} - R^2_{CD})}{3}\right] + \left[\frac{(R^2_{AB} - R^2_{B}) + (R^2_{AC} - R^2_{C}) + (R^2_{AD} - R^2_{D})}{3}\right] + R^2_A\right\}}{4}$$

$$Shapley\ Value\ (B) = \frac{\left\{(R^2_{ABCD} - R^2_{ACD}) + \left[\frac{(R^2_{ABC} - R^2_{AC}) + (R^2_{ABD} - R^2_{AD}) + (R^2_{BCD} - R^2_{CD})}{3}\right] + \left[\frac{(R^2_{AB} - R^2_{A}) + (R^2_{BC} - R^2_{B}) + (R^2_{BD} - R^2_{D})}{3}\right] + R^2_B\right\}}{4}$$

$$Shapley\ Value\ (C) = \frac{\left\{(R^2_{ABCD} - R^2_{ABD}) + \left[\frac{(R^2_{ABC} - R^2_{AB}) + (R^2_{ACD} - R^2_{AD}) + (R^2_{BCD} - R^2_{BD})}{3}\right] + \left[\frac{(R^2_{AC} - R^2_{A}) + (R^2_{BC} - R^2_{B}) + (R^2_{CD} - R^2_{D})}{3}\right] + R^2_C\right\}}{4}$$

$$Shapley\ Value\ (D) = \frac{\left\{(R^2_{ABCD} - R^2_{ABC}) + \left[\frac{(R^2_{ABD} - R^2_{AB}) + (R^2_{ACD} - R^2_{AC}) + (R^2_{BCD} - R^2_{BC})}{3}\right] + \left[\frac{(R^2_{AD} - R^2_{A}) + (R^2_{BD} - R^2_{B}) + (R^2_{CD} - R^2_{C})}{3}\right] + R^2_D\right\}}{4}$$

Where R^2_A is the corresponding fit value when the model contains only the regressor A, R^2_B when the model contains only B, and so on.

The first and middle terms for each equation accounts for the contribution of variable X when other variables are added in the model. All these marginal effects are then averaged for each for each variable, and that constitutes the corresponding Shapley values. The variable having the maximum Shapley value is considered as the key driver for that specific dataset under a certain scenario.

Within one scenario, the process is done 400 times (as there are 400 datasets for each scenario). The variable that has the maximum count (i.e. most frequent key driver for the 400 datasets) is the one Shapley regression considers as the key driver for this scenario, while the one with the second highest count is the second driver. The frequency of each question being identified as the key driver and second driver will also be taken.

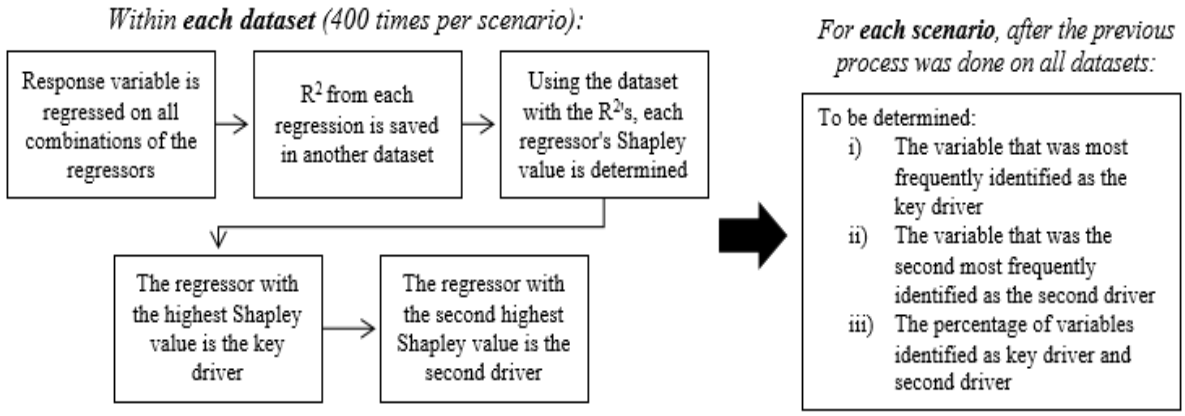

Figure 3: A flowchart of the steps to be taken in KDA using Shapley

Comparison of the Results of the Two Methods and Other Analyses

The two methods will be done simultaneously for each of the data sets simulated (scenarios), according to the following general algorithm:

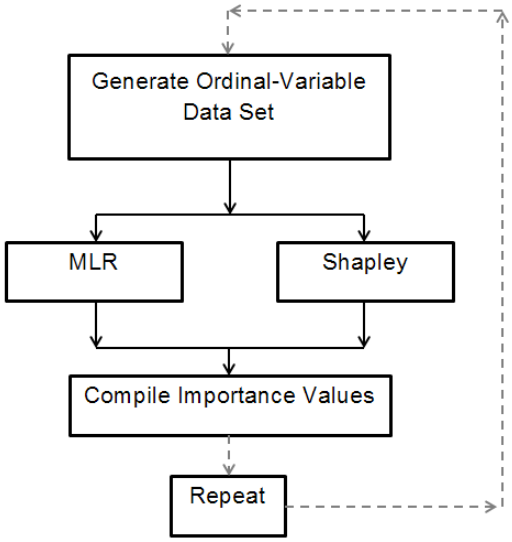

Figure 4: Overall algorithm to be used in simulations

Now that the summary of identified questions as key and second drivers for each of the methods were outputted, the researchers will compare them. For each scenario, the researchers will determine which method is more appropriate by choosing the one that got the correct key and second drivers the most times. In the case that the results of the two methods are almost equal, the MLR method will be chosen due to the simplicity of its application. In addition, the researchers will inspect the behavior of the estimated coefficients of the MLRs and compare them to their supposed weights (assigned in the simulation) to see how closely they were able to be estimated.

Results and Discussions

In simulating multicollinearity among the *p* questions, the following marginal PMFs were used in the simulation:

Table 2: Marginal PMFs defined in simulating Likert variables

X	1	2	3	4	5
P{X1=X}	0.20	0.20	0.20	0.20	0.20
P{X2=X}	0.10	0.20	0.40	0.20	0.10
P{X3=X}	0.35	0.35	0.10	0.10	0.10
P{X4=X}	0.20	0.30	0.30	0.10	0.10

The rationale behind the above set marginal PMFs is that several scenarios of distributions of answers are simulated per question. X1 represents a question regarding an attribute to which the respondents are indifferent, thus the probabilities of each of the possible responses are equal. On the other hand, X2 represents a question to which the respondents tend to be neutral, thus the probabilities are symmetrically distributed around the middle value 3. X3 represents a question to which the respondents are generally negative to, thus the probabilities are distributed skewed to the right. X4 represents a question to which the respondents are negative to neutral to, thus although still being skewed to the right is considerably distributed closer towards the center than X3.

Likewise, the following target correlation matrix was calibrated in simulating the four ordinal variables using the Mean Mapping Method by Kaiser et. al.

Table 3: Table of target correlations between the four simulated variables

	X1	X2	X3	X4
X1	1	0.80	0.80	0.75
X2		1	0.90	0.80
X3			1	0.80
X4				1

For all cases, $p = 4$ ordinal variables will be simulated, simply omitting one of them from the regression model during MLR estimation to simulate the scenario that a significant variable influencing the dependent Y has not been included by the researchers performing the survey. The above correlation matrix will be used in the scenarios where multicollinearity is present. For the case where no multicollinearity is present, an identity matrix will be used as the target correlation matrix.

As a prerequisite step in the study, the Mean Mapping Algorithm method was tested for each of the 400 datasets simulated to check how far the resulting correlations were from the target. For the correlation between X1 and X2, a mean absolute error of 0.0258. Throughout the 400 simulated data sets, the resulting correlation was only 0.03 units above or below the target, suggesting a good amount of intended correlation retained in simulation.

For all scenarios under case 1, the questions have the following weights, question 1: 0.05, question 2: 0.15, question 3: 0.20, and question 4: 0.60. Question 4 is the clear key driver while question 3 is the second. On the other hand, for scenarios under case 2, the weights are the following, question 1: 0.10, question 2: 0.25, question 3: 0.30, and question 4: 0.35. Lastly, for all scenarios under case 3, question 1: 0.20, question 2: 0.23, question 3: 0.27, question 4: 0.30. In here, questions have almost equal influence on the Y, making the key driver question 4 and second driver question 3 difficult to detect.

In the scenarios with an omitted question, question 3 will be omitted, hence the second driver will be question 2. The table of percentages of questions identified as the two main drivers will be presented. To analyze the behavior of the two methods' key driver determinants throughout the 400 datasets, the scatterplot of each of them against that regression's R^2 are presented for each scenario.

Case 1: Main Drivers[1] are Strongly Evident

Scenario: Multicollinear, no omitted variables

The following discussion summarizes the result of MLR and Shapley in finding the key driver for each of the 400 simulated datasets under the scenario that the four regressors are multicollinear and no significant question was omitted. Each batch of driver importances estimated for one simulated dataset is considered 1 instance. The percentage of instances in which a question was counted as the key driver (highest importance) or second driver (second-highest importance) are presented in Table 4.

Shapley regression correctly classified question 4 as the key driver 3.5 percentage points of the time more than MLR was able to, and question 3 as the second driver 4.5 percentage points of the time more. For this scenario, Shapley regression more accurately determined the key driver than MLR although only by fractional difference.

Table 4: Percentage of instances a question was identified as a main driver

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.75%	1.25%	6.25%	5.25%
2	0.75%	0.50%	14.25%	16.25%
3	23.25%	19.50%	57.25%	61.75%
4	75.25%	78.75%	22.25%	16.75%

Meanwhile, figures 5 and 6 present a scatterplot of the driver importance values computed for the key drivers using MLR and Shapley against the R^2 fits of their corresponding models.

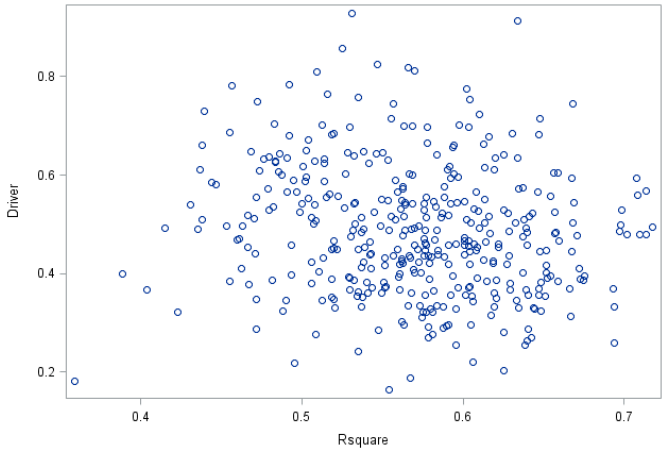

Figure 5: Identified key driver's importance (MLR) vs Fit Performance

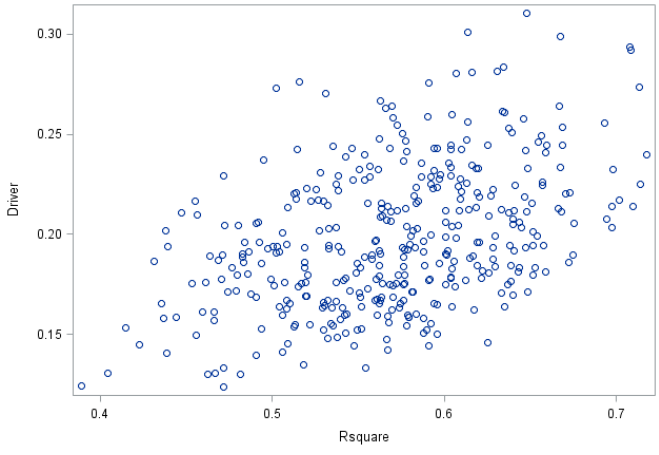

Figure 6: Identified key driver's Shapley value vs Fit Performance

[1] Henceforth used to refer collectively to the key and second drivers.

The way the points are exploded erratically suggest an instability in the MLR *importance* estimates. Regardless of how good the regression model fits the simulated data, the *importance* values do not seem to exhibit any stability. It is evident in the scatterplot that even when the R^2 fit is good, the key driver's *importance* value can be very low or very high, without a distinguishable pattern.

On the other hand, Figure 6 shows an increasing trend in the Shapley value when the R^2 fit is improving, although this result is slightly expected considering that the Shapley value is a function of the all-possible-regression R^2 values. But because of the presence of multicollinearity, even the Shapley values display some level of erraticness, which must be due to high multicollinearity levels blowing up the R^2 values for the permutation models used in Shapley.

Scenario: Multicollinear, Question 3 omitted

Table 5: Percentage of instances a question was identified as a main driver

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.75%	1.75%	29.25%	19.00%
2	7.75%	6.00%	63.00%	74.00%
3	-	-	-	-
4	91.50%	92.25%	7.75%	7.00%

Shapley regression identified the correct key driver by a mere 0.75 percentage points more than the MLR and the correct second driver by a much larger 11 percentage points. Shapley remains the better method in identifying drivers, although for the case of the key driver, the small advantage of Shapley over MLR might seem too small considering the tediousness of the computation process involved.

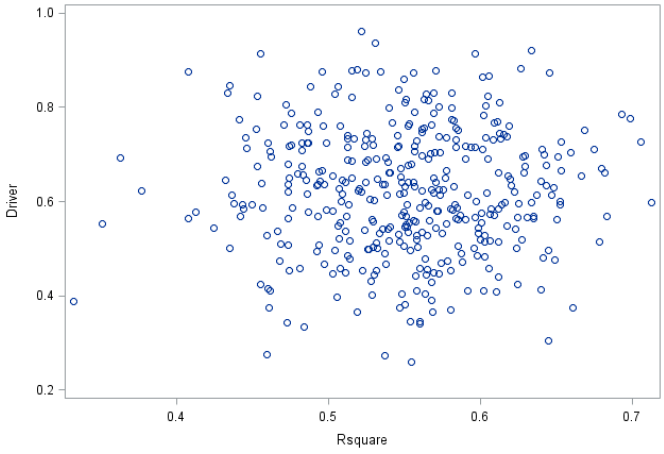

Figure 7: Identified key driver's importance (MLR) vs Fit Performance

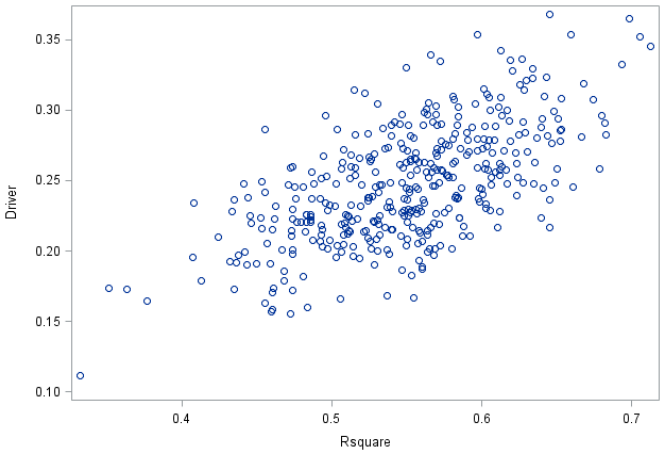

Figure 8: Identified key driver's Shapley value vs Fit Performance

The same patterns as when the regressors were complete are observed with both methods when the second driver is omitted from the model. MLR estimates are erratic and wildly scattered even when the R^2 fit of the model to the data is at its highest. Meanwhile, Shapley retains the increasing linear trend, with slightly higher values that are now more close together.

Scenario: Non-collinear, no omitted questions

Both methods were able to identify the correct key driver 100 percent of the time, however, MLR will be chosen due to its relative ease of computation. Shapley regression, on the other hand, identified the correct second driver more than the MLR by 6.75 percentage points. Again, the advantage is very low compared to how tedious the process involved is for Shapley regression. The behavior of the estimates, however, shows the same pattern as in the multicollinear case, although here there is a slight increasing trend distinguishable from the scatter plot for MLR. The estimates remain to be to some degree erratic.

Table 6: Percentage of instances a question was identified as a main driver

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.00%	0.00%	4.75%	8.00%
2	0.00%	0.00%	32.50%	22.50%
3	0.00%	0.00%	62.75%	69.50%
4	100.00%	100.00%	0.00%	0.00%

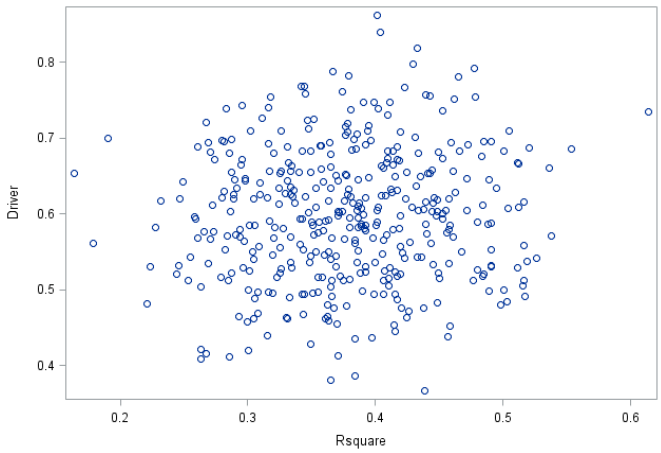

Figure 9: Identified key driver's importance (MLR) vs Fit Performance

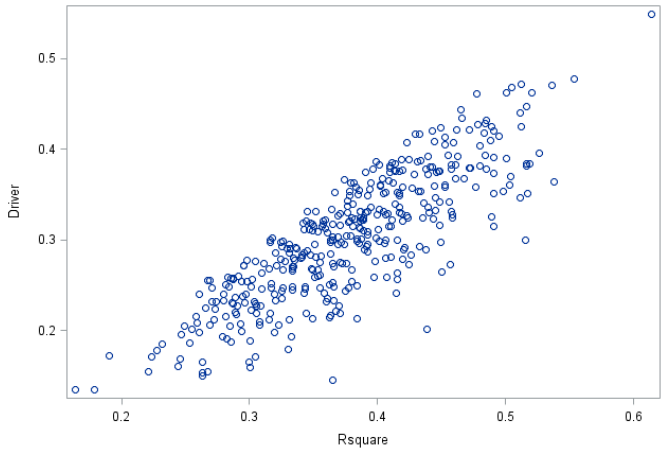

Figure 10: Identified key driver's Shapley value vs Fit Performance

Now that the questions are non-collinear, the range of the R2's is lower but the importance still erratic. The Shapley values on the other hand, are now a lot more precise and have an even clearer increasing linear trend.

Scenario: Non-collinear, Question 3 omitted

Table 7: Percentage of instances a question was identified as a main driver

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.00%	0.00%	25.75%	32.75%
2	0.00%	0.00%	74.25%	67.25%
3	-	-	-	-
4	100.00%	100.00%	0.00%	0.00%

Both methods were able to identify the correct key driver. Again, MLR is chosen over Shapley regression. Moreover, MLR works better than Shapley as it correctly identified the second driver 7 percentage points higher than Shapley. While Shapley proved to be more powerful than MLR in the multicollinear scenarios, MLR seems to be more preferable for the case of non-collinear explanatory variables.

The Importance vs R2 behavior remains similar to all the other cases, however the Shapley values obtained almost perfectly increase with the R2. See Figures 11 and 12 below:

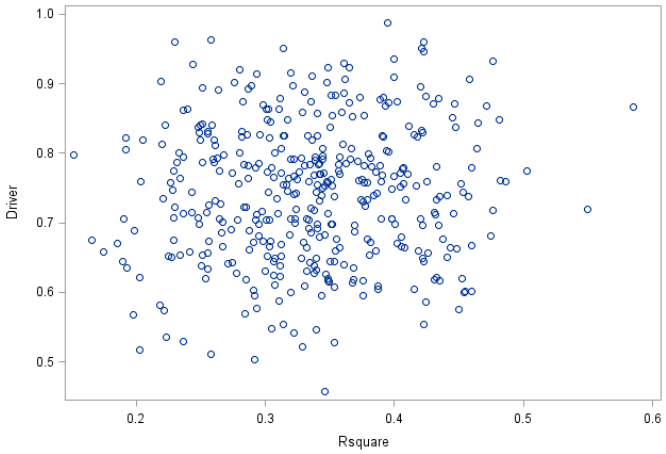

Figure 9: Identified key driver's importance (MLR) vs Fit Performance

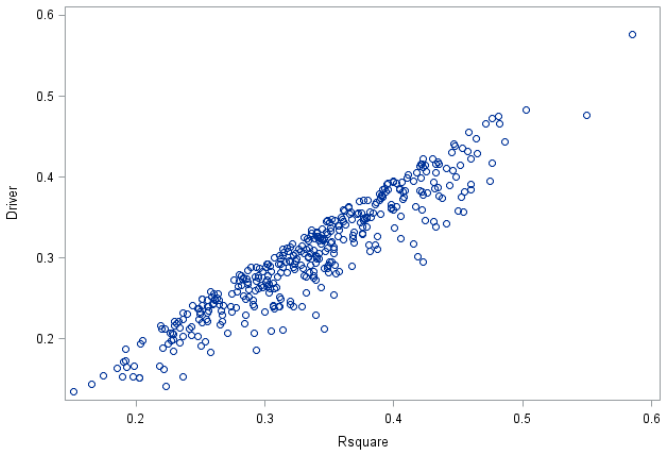

Figure 10: Identified key driver's Shapley value vs Fit Performance

Case 2: Main Drivers Fairly Evident

With a hypothetical model using regression weights in which the main drivers are somewhat indiscernible, both techniques failed in identifying the key and second drivers when multicollinearity is present and no question was omitted. However, it can be seen that both techniques might have also switched the ranks of question 3 and question 4, which is clearly evident for the case of MLR. Compare the results under high multicollinearity (Table 8) and under perfect non-collinearity (Table 9). Highlighted in cyan are percentages that are highest for each technique, indicating the question number it identified as the key driver/second driver. Highlighted in yellow are percentages that are higher between the two techniques, indicating the technique that performed better. Note that when there is perfect independence between the four variables, the correct main drivers were again identified by both MLR and Shapley.

Table 8: Percentage of instances a question was identified as a main driver under case 2, collinear with no omitted questions

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	2.00%	3.50%	11.00%	6.50%
2	8.00%	5.75%	14.00%	33.75%
3	62.50%	63.25%	**23.00%**	**26.25%**
4	**27.50%**	**27.50%**	52.00%	33.50%

Table 9: Percentage of instances a question was identified as a main driver under case 2, non-collinear with no omitted questions

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.00%	1.00%	4.25%	6.00%
2	10.50%	6.25%	25.25%	19.75%
3	35.00%	43.00%	39.50%	39.50%
4	54.50%	49.75%	31.00%	34.75%

Both techniques are now being confused that they have been identifying question 3 as the key driver when it should have been question 4 (and vice versa) when the variables are multcollinear. Comparing the two techniques performance in identifying the correct main drivers, both equally performed in determining the key driver at 27.50%, while Shapley identified the second driver 3.25 percentage points higher than the MLR. This confusion is not observed under non-collinearity, although MLR performed better than Shapley with 4.75 percentage points higher in determining the key driver and just the same as in determining the second driver. This might indicate that MLR indeed performs better than Shapley regression under a “no-problem” scenario (i.e. no multicollinearity and no omitted questions).

The same results are observed when question 3 has been omitted. With multicollinearity, Shapley regression identified the correct key driver by a mere 0.75 percentage points more than the MLR and the correct second driver by a much larger 11 percentage points (see table A1 in Appendix A). Meanwhile, under non-collinearity, both methods were able to identify the correct key driver although MLR works better than Shapley as it correctly identified the second driver 7 percentage points higher than Shapley. While Shapley proved to be more powerful than MLR in the multicollinear scenarios, MLR seems to be more preferable for the case of non-collinear explanatory variables (see table A2 in Appendix A).

The patterns in the importance vs fit plots are similar to those seen in the first case: the importance estimates using the MLR erratic, while Shapley values exhibit an increasing linear trend. This still shows that the better the fit or the higher the R2 means the greater chance Shapley results will be reliable. However, in this case, the techniques experience problems in correctly identifying the main drivers; specifically, it switches the ranks of the key driver and second driver. See figures A1 to A4 in Appendix A for the scatterplots.

Case 3: Main Drivers Indiscernible

Main drivers were misidentified for all scenarios, indicating that both techniques failed. Highlighted in cyan are percentages that are highest for each technique, indicating the question number that it identified as the key driver/second driver.

Without omitting any questions, under multicollinearity both MLR and Shapley regression were incapable of correctly identifying the key and second driver. Both technique identified question 3 as the key driver. On the other hand, MLR identified question 4 as the key driver, while Shapley identified question 2. With this, it can be seen in MLR that the identification of the two main drivers were switched (i.e. question 3 was identified as the key driver when it's supposed to be question 4, while question 4 was identified as the second driver when it's supposed to be question 3). Shapley, on the other hand, failed at determining the key and second driver. Ignoring that both have failed, MLR identified the key driver 2.75 percentage points higher than Shapley but Shapley identified the second driver 5.5 percentage points higher than MLR. In MLR, it can be seen that the identification of the two main drivers were switched. However, when the regressors are non-collinear both methods correctly identified the drivers. MLR was able to correctly identify the key and second drivers than Shapley by 4 percentage points and 3.75 percentage points, respectively. Shapley failed at identifying the second driver that may be possibly because of switching again both the third and fourth questions as the key drivers; MLR was able to find the correct second driver. These results are summarized in tables 10 and 11 respectively.

Table 10: Percentage of instances a question was identified as a main driver under case 2, collinear with no omitted questions

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	7.00%	11.75%	27.50%	19.25%
2	8.00%	5.25%	11.25%	31.75%
3	62.00%	62.75%	**18.25%**	**23.75%**
4	**23.00%**	**20.25%**	43.00%	25.25%

Table 11: Percentage of instances a question was identified as a main driver under case 2, non-collinear with no omitted questions

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	9.50%	15.50%	13.25%	21.50%
2	13.75%	8.00%	20.75%	16.25%
3	34.50%	38.25%	**33.50%**	**29.75%**
4	42.25%	38.25%	32.50%	32.50%

The same results are obtained when question 3 has been omitted from the model. Under collinearity, both techniques identified question 2 as the key driver when it is only the second driver. With question 3 omitted, the second driver is now question 2, but none of the techniques were able to determine the second driver correctly. It can be seen, however, that Shapley had switched the identification of the key and second driver. Disregarding that both have failed, MLR actually got the correct key driver 3.25 percentage points higher than Shapley but Shapley got the second driver 7.25 percentage points higher than MLR. And under non-collinearity, MLR got the key and second drivers higher than Shapley by 4.25 percentage points and 9.50

percentage points, respectively. With no signs of switching the identification of key and second drivers, it is clear that Shapley has failed, falling behind MLR by at least 5 percentage points. These results are summarized in tables A3 and A4

The same driver importance vs fit patterns are observed in all four scenarios under case 3 as in the previous cases. The scatterplots can be found in figures A5 to A8.

On the behavior of MLR Estimates

From Case 1: $Y_i = 0.05\ X1_i + 0.15\ X2_i + 0.20\ X3_i + 0.60\ X4_i + \varepsilon_i$

Table 12: MLR Estimates behavior under multicollinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.0282638	0.1118664	0.0172678	0.0392599
X2	-0.0124018	0.1798682	-0.0300822	0.0052786
X3	0.2969792	0.1663545	0.2806272	0.3133312
X4	0.4758095	0.1358109	0.4624598	0.4891592

Table 13: MLR Estimates behavior under non-collinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.0472533	0.0626306	0.0410970	0.0534097
X2	0.1164651	0.0831275	0.1082939	0.1246362
X3	0.1655813	0.0718766	0.1585161	0.1726465
X4	0.5088389	0.0761798	0.5013507	0.5163270

The values are observed to be closer to the actual regression weights used in the simulation in the case of no collinearity between regressors. The standard deviations are also smaller, indicating greater stability in the estimates, whereas those in the multicollinear case have very high variability.

These results can be viewed in reference to the Fit vs *Importance* scatterplots discussed earlier in the paper, where the estimates were blown up erratically in the multicollinear case, and more stable in the noncollinear case - the trend becoming more apparent although the variability remained considerably wild. This also explains how MLR was able to approach and even better the performance of Shapley regression in the noncollinear setting.

It is now also apparent why Shapley regression is preferable when multicollinearity exists among the variables. In this scenario, the beta estimates tend to be overblown, thus any measure depending on them, though are not exactly inaccurate, would be wild and the probability of getting an incorrect key driver high (almost 25 percent for the case of multicollinear, no omitted questions). Shapley solves this by foregoing use of the beta estimates themselves and using instead an alternative measure of the *importance* using the R^2 fit. It must be noted that Shapley itself also uses OLS regression, which is why MLR begins to mirror its results when the beta estimates too begin to stabilize.

This stability, however, will always remain incomplete, as has been discussed in related literature, due to the data being Likert-type (or ordinal in general). Variability is even greatly exaggerated by the discreteness of the data points, and the fact that the number of mass points

are too small, that a lot of variation is held within one step, instead of being distributed along an infinitely long continuous scale as is the ideal case when the regressors and the dependent variable are all normally distributed.

From Case 2: $Y_i = 0.10\ X1_i + 0.25\ X2_i + 0.30\ X3_i + 0.35\ X4_i + \varepsilon_i$

Table 14: MLR Estimates behavior under multicollinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.0676911	0.1094237	0.0569351	0.0784470
X2	0.0565236	0.1827198	0.0385629	0.0744843
X3	0.3895534	0.1671959	0.3731186	0.4059881
X4	0.2808217	0.1376425	0.2672919	0.2943514

Table 15: MLR Estimates behavior under non-collinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.0910439	0.0658451	0.0845716	0.0975163
X2	0.2020139	0.0824509	0.1939093	0.2101185
X3	0.2632157	0.0761245	0.2557329	0.2706985
X4	0.3001297	0.0795333	0.2923119	0.3079475

Under the presence of multicollinearity, MLR estimates of *importance* have given more weight to question 3 at almost 0.40, followed by question 4 at 0.28. This explains why MLR kept on switching the identification of the key driver and main driver. Also, as seen in the confidence intervals, the regression coefficients provided by the MLR technique are also not anywhere near the true weights of the question.

As for the scenario of non-collinearity, none of the true weights were still captured by the confidence intervals (despite X1 being very near). However, in this scenario, it can now be seen that the weights have been appropriately distributed among the four questions, with question 4 having the highest average *importance* followed by question 3. Despite the *Non-collinear, No Omitted* scenario being the "no-problem" scenario, similar results can also be seen even in *Non-collinear, Omitted* scenario. This is because OLS beta-coefficients are only affected by omitted variable bias only when the omitted variable is correlated with the regressors (i.e. OLS beta-coefficients remain unbiased and consistent). In Shapley's case, however, an omission of a relevant variable causes R^2 (and therefore Shapley values) to destabilize. This might explain why MLR is consistently performing just as well or even better than Shapley under scenarios involving non-collinearity,

Similar in case 1 wherein the *Importance* vs. R^2 graphs involving non-multicollinearity seem to start showing an increasing trend (though still considerably erratic), the same-scenario graphs under case 2 also exhibit the same pattern. This is because as mentioned earlier, the *importance* estimates under non-collinearity, despite being far from the true values, still possessed the correct order of weights (increasing trend). Their standard deviations are also low, implying accurate *importance* estimates. With this, MLR was able to correctly identify the main drivers even better than Shapley.

From Case 3: $Y_i = 0.20\ X1_i + 0.23\ X2_i + 0.27\ X3_i + 0.30\ X4_i + \varepsilon_i$

Table 16: MLR Estimates behavior under multicollinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.1507032	0.1135424	0.1395424	0.1618641
X2	0.0426654	0.1817157	0.0248034	0.0605273
X3	0.3648335	0.1669064	0.3484272	0.3812398
X4	0.2387454	0.1357222	0.2254044	0.2520864

Table 17: MLR Estimates behavior under non-collinearity and without omitted questions

Variable	Mean	Std. Dev	Lower 95% CL (Mean)	Upper 95% CL (Mean)
X1	0.1753622	0.0630472	0.1691649	0.1815595
X2	0.1910458	0.0809131	0.1830924	0.1989993
X3	0.2401289	0.0742449	0.2328309	0.2474269
X4	0.2606342	0.0787232	0.2528960	0.2683724

Similar in case 2, the magnitudes were switched for question 3 and question 4. This is evident in the results where MLR (now, including Shapley) switches the identification of the main drivers. Given that MLR performs bad under multicollinearity, it further worsens when drivers are almost as influential as each other. Not only that it switches the places of the key driver and second driver, it gives *importance* estimates that are arbitrarily far from the true weights. This is clear as their standard deviation is consistently high and the true weights lie far outside the confidence intervals.

While the regression coefficients lie still far away from the true weights under non-collinear scenario, they now exhibit an increasing linear trend. The low standard deviations also suggest the accuracy of the estimates. This is why despite the bias being large, MLR was able to correctly identify the key driver and second driver even better than the Shapley driver. This further stresses out how MLR performs well under non-collinearity, regardless of 1) an omission of relevant variable and/or 2) indiscernibility of the main drivers.

Conclusions

It has been shown that Shapley performed better than MLR in scenarios considered in the present paper wherein the key driver is obvious based on the regression weights, but even then the advantage is too low to justify its complicated computation process. MLR, although it produced erratic and highly deviating values for the questions still performed almost as well as Shapley regression. However, this same result was not seen in cases wherein the main drivers are to some degree indiscernible, as the erratic behaviour of linear regression estimates (on which both MLR and Shapley are based) tend to overestimate some and underestimate others, to the point that it confuses between the key and second drivers.

Recommendations

Future analysis of this same topic might include varying levels of multicollinearity for each case, to explore how the MLR estimates stabilize as the level of collinearity between each variable exists. At the same time, future studies might consider the incorporation of various other KDA analysis methodologies, such as Kruskall's Driver Analysis and Random Forests. Using mixed continuous and categorical variables may also be explored.

Another possible setup that might be recalibrated from this simulation study is the algorithm used to simulate the data. The Mean Mapping Method by Kaiser et al uses the multivariate normal distribution as a starting point and then transforms the continuous scale into discrete ordinal using the marginal PMFs for the target variables and their corresponding "mappings" from the univariate standard normal distribution. In this respect, the resulting data follows to some degree the behavior of MVN-distributed random variables.

Other simulation techniques that do not make use of MVN distribution might lead to different results. Another powerful simulation algorithm recommended in this regard is presented in Demirtas (2006), which uses a bivariate uniform normal distribution as starting point. Various other methods presented in Biswas (2004) may also be attempted, which use non-distributional algorithms.

References

Allen, Elaine and Seaman, Christopher (2007). "Likert Scales and Data Analyses". *Quality Progress*. pp. 64–65.

Biswas, Atanu (2004). "Generating correlated ordinal categorical random samples." Statistics & Probability Letters 70 (2004) 25–35.

Demirtas, H. (2006), "A Method for Multivariate Ordinal Data Generation Given Marginal Distributions and Correlations," *Journal of Statistical Computation and Simulation*, 76, 1017–1025.

Driver (Importance) Analysis. 3 August 2016. In *Q*. Retrieved from http://wiki.q-researchsoftware.com/wiki/Driver_(Importance)_Analysis

Kaiser, S., Träger, D., and Leisch, F. (2011), Generating Correlated Ordinal Random Values, Technical report, University of Munich, Department of Statistics. URL http://epub.ub.uni-muenchen.de/12157/

Kruskal, W. (1987), "Relative Importance by Averaging over Orderings," *The American Statistician*, 41, 6-10.

Labovitz, S (1967). "Some observations on measurement and statistics". *Social Forces*. 46: 151–160. doi:10.2307/2574595

Likert, Rensis (1932). "A technique for the measurement of attitudes." *Archives of Psychology*, 1932, Vol 140, No. 55. Cited in Allen et al (2007)

Munshi, Jamal (2014). "A Method for Constructing Likert Scales". ssrn.com. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2419366

Osborne, Jason W. and Waters, Elaine (2002). "Four Assumptions Of Multiple Regression That Researchers Should Always Test." *Practical Assessment, Research &*

Evaluation, Vol.8, No.2. ISSN: 1531-7714. URL: http://pareonline.net/getvn.asp?n=2&v=8

Sambandam, R. (2001), "Survey of Analysis Methods Part I: Key Driver Analysis". trchome.com. URL: http://www.trchome.com/component/content/article/66-market-research-knowledge/published-articles/206-survey-of-analysis-methods-key-driver-analysis

Wicklin, R. (2013), *Simulating Data with SAS*, Cary, NC: SAS Institute Inc.

Winship, Christopher and Robert Mare (1984). "Regression Models With Ordinal Variables." *American Sociological Review*, Vol. 49 (August:512-525). URL: http://scholar.harvard.edu/files/cwinship/files/asr_1984.pdf

Miscellaneous:

Hansa Marketing Services, Blog (2014) "Brand Assessment Tools: Measuring Relative Importance with Shapley Value Regression." Accessed 25 October, 2016. URL:hansamarketing.com/blog/bid/238057/Brand-Assessment-Tools-Measuring-Relative-Importance-with-Shapley-Value-Regression

Appendix A: Other summary tables and figures

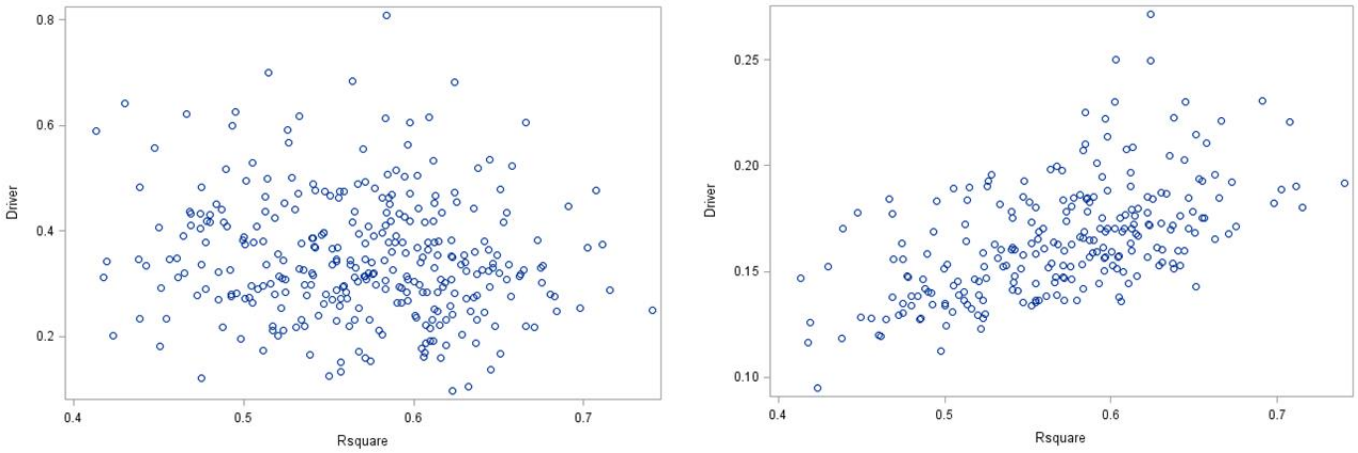

Figure A1: Importance vs fit performance plots for MLR (left) and Shapley (right) with multicollinear regressors and no question omitted under case 2

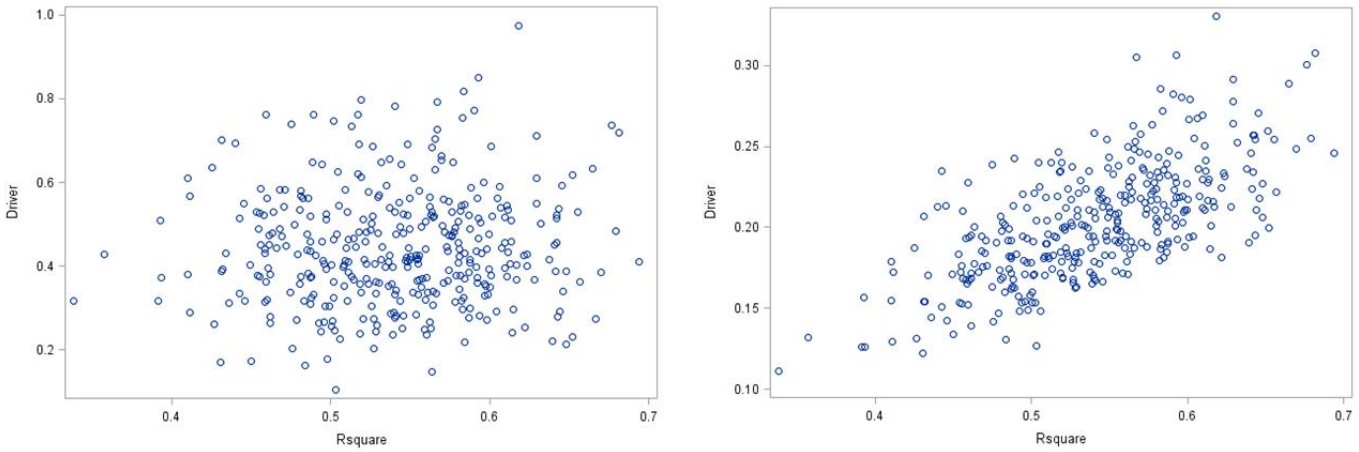

Figure A2: Importance vs fit performance plots for MLR (left) and Shapley (right) with multicollinear regressors and one question omitted under case 2

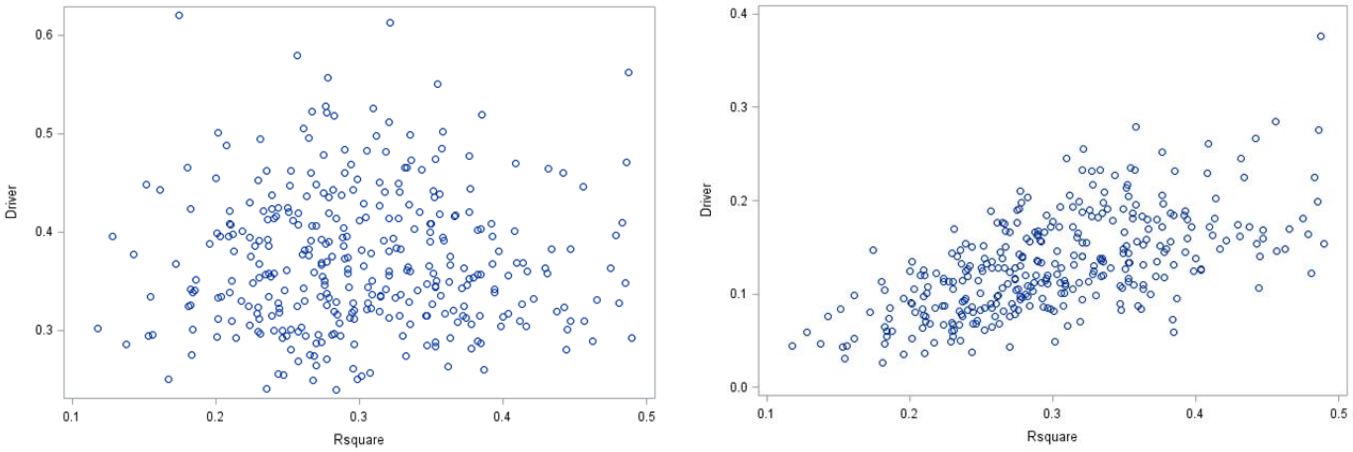

Figure A3: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and no question omitted under case 2

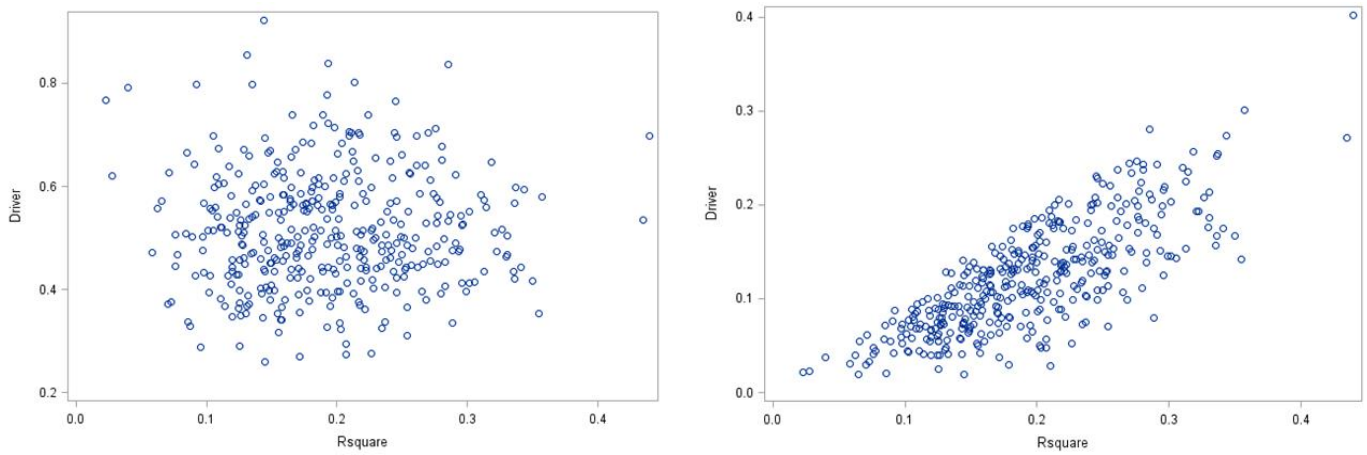

Figure A4: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and one question omitted under case 2

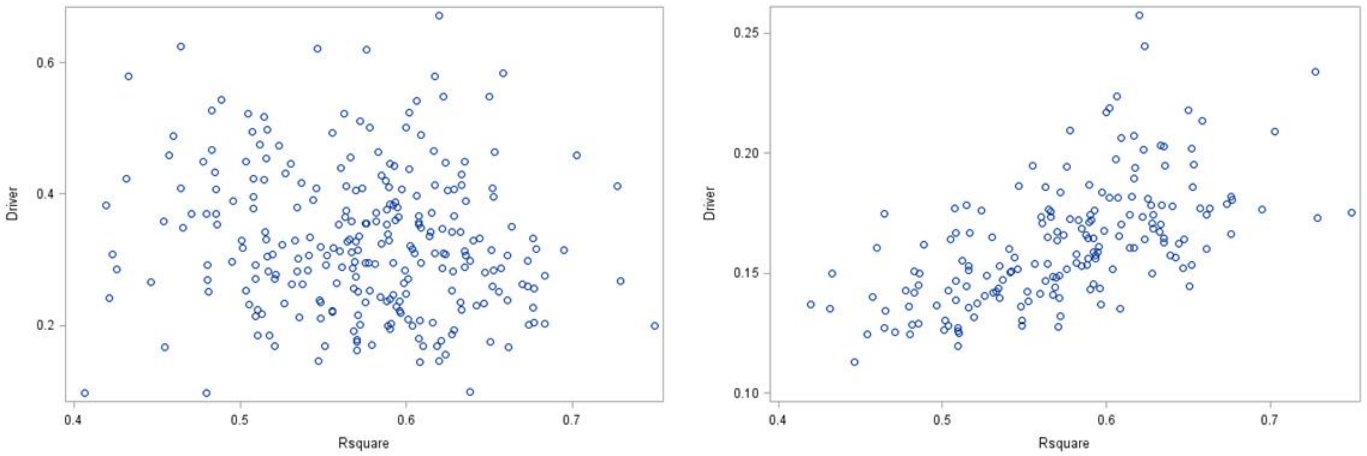

Figure A5: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and one question omitted under case 3

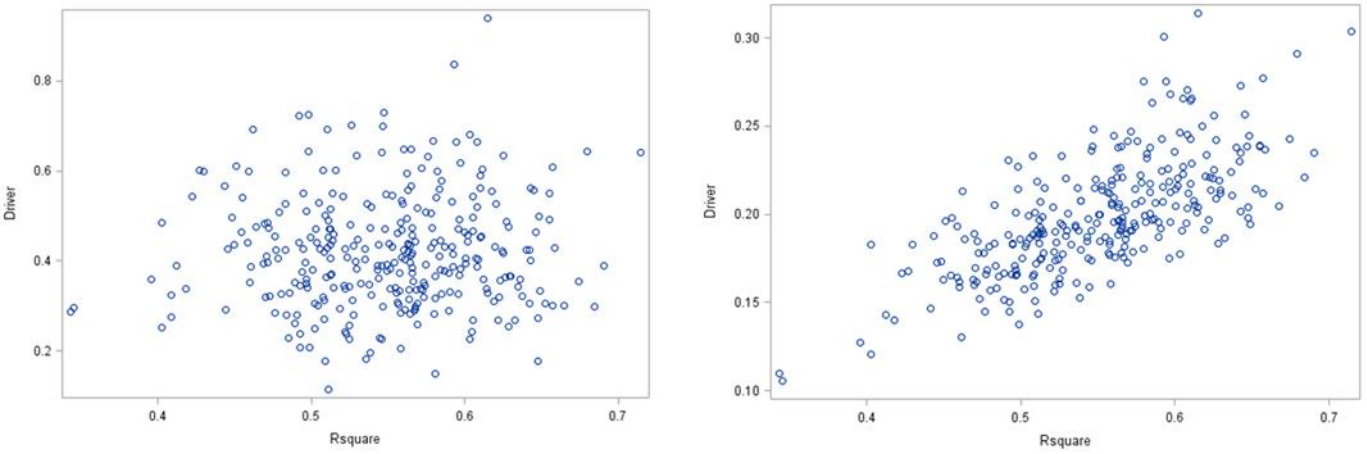

Figure A6: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and one question omitted under case 3

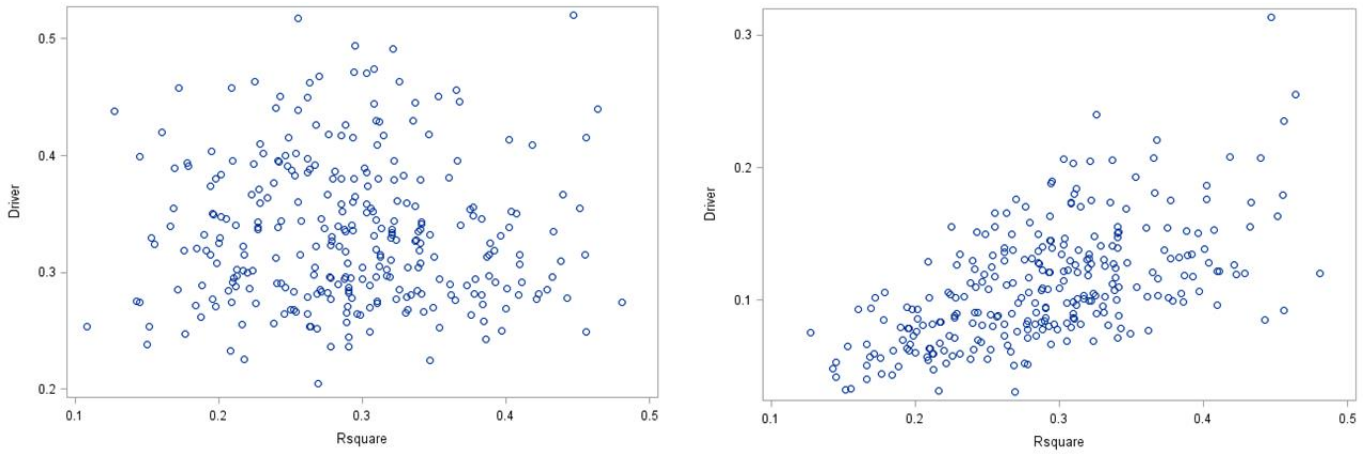

Figure A7: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and one question omitted under case 3

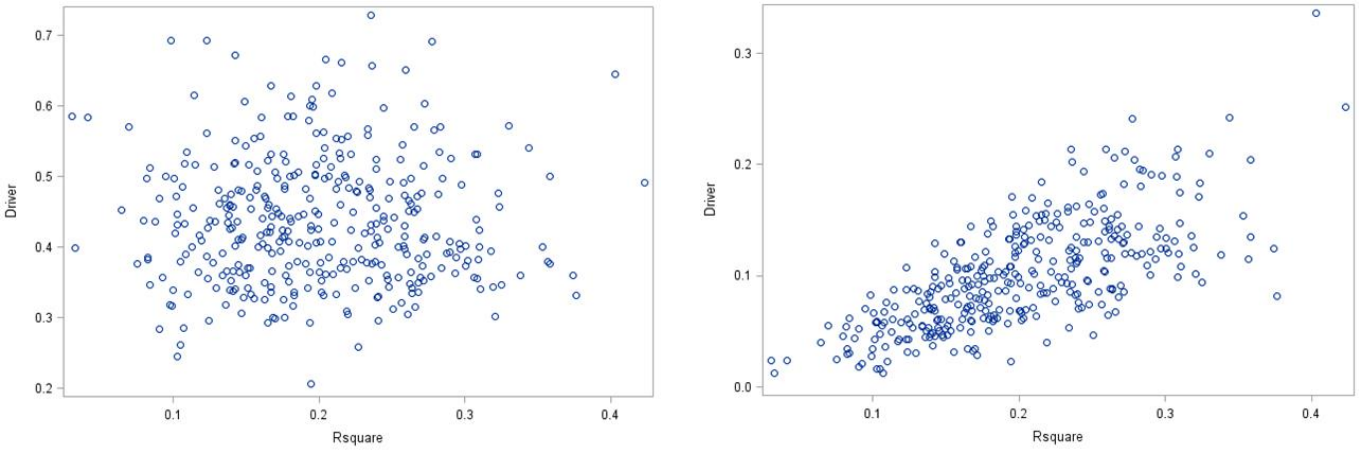

Figure A8: Importance vs fit performance plots for MLR (left) and Shapley (right) with non-collinear regressors and one question omitted under case 3

Table A1: Percentage of instances a question was identified as a main driver under case 2, collinear with one omitted questions under case 2

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.75%	1.75%	29.25%	19.00%
2	7.75%	6.00%	63.00%	74.00%
3	-	-	-	-
4	91.50%	92.25%	7.75%	7.00%

Table A2: Percentage of instances a question was identified as a main driver under case 2, non-collinear with one omitted questions under case 2

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	0.00%	0.00%	25.75%	32.75%
2	0.00%	0.00%	74.25%	67.25%
3	-	-	-	-
4	100.00%	100.00%	0.00%	0.00%

Table A3: Percentage of instances a question was identified as a main driver under case 2, collinear with one omitted questions under case 3

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	12.25%	22.00%	37.25%	30.75%
2	46.50%	40.00%	**27.00%**	**34.25%**
3	-	-	-	-
4	**41.25%**	**38.00%**	35.75%	35.00%

Table A4: Percentage of instances a question was identified as a main driver under case 2, non-collinear with one omitted questions under case 3

	Key Driver		Second Driver	
Question	**MLR**	**Shapley**	**MLR**	**Shapley**
1	14.50%	24.75%	33.50%	40.50%
2	22.00%	16.00%	**39.50%**	**30.00%**
3	-	-	-	-
4	63.50%	59.25%	27.00%	29.50%

Appendix B: SAS codes used

```
/*************************************************************
CODES PREPARED BY
BENABAYE, ISABELLA
CABRAL, ANNE LOUISE
DAYTA, DOMINIC
DONATO, PATRICIA ROSE

PRESENTED TO
SIR MICHAEL VAN SUPRANES

AS PARTIAL FULFILLMENT OF REQUIREMENTS IN
STATISTICS 196.1, ADVANCED STATISTICAL COMPUTING

Initial Caveats: Algorithm used for the simulation of
ordinal-level variables from multivariate normal variates
were derived from Wicklin (2013).
*************************************************************/


/*************************************************************
The following block of code concerns with the generation of
p likert variables based on a defined PMF and correlation
structure.
*************************************************************/
%macro ordsimulate(seed,Corr);

        *the functions OrdN, OrdVar, OrdCDF, Expand2Grid,
        OrdFindRoot, OrdQuant, OrdMVCorr, and RandMVOrdinal
        are functions defined in Wicklin(2013);

        proc iml;
```

```
/* OrdN: number of values for each variable */
start OrdN(P);
        return( countn(P, "col") );
finish;

/* OrdMean: Expected value for each variable is Sigma_i (i*p[i])*/
start OrdMean(P);
        x = T(1:nrow(P)); /* values of ordinal vars */
        return( (x#P)[+,] ); /* expected values E(X) */
finish;

/* OrdVar: variance for each variable */
start OrdVar(P);
        d = ncol(P); m = OrdMean(P);
        x = T(1:nrow(P)); /* values of ordinal vars */
        var = j(1, d, 0);
        do i = 1 to d;
                var[i] = sum( (x - m[i])##2 # P[,i] ); /* defn of variance */
        end;
        return( var );
finish;

/* OrdCDF: Given PMF, compute CDF = cusum(PDF) */
start OrdCDF(P);
        cdf = j(nrow(P), ncol(P)); /* cumulative probabilities */
        do i = 1 to ncol(P);
                cdf[,i] = cusum(P[,i]);
        end;
        return( choose(P=., ., cdf) ); /* missing vals for short cols */
finish;

/* Function that returns ordered pairs on a uniform grid of points.
```

```
Return value is an (Nx*Ny x 2) matrix */
start Expand2DGrid( _x, _y );
    x = colvec(_x); y = colvec(_y);
    Nx = nrow(x); Ny = nrow(y);
    x = repeat(x, Ny);
    y = shape( repeat(y, 1, Nx), 0, 1 );
    return ( x || y );
finish;


/* OrdQuant: Compute normal quantiles for CDF(P) */
start OrdQuant(P);
    N = OrdN(P);
    CDF = OrdCDF(P);

    /* QUANTILE function in SAS/IML 12.1 does not accept 1 as
parameter */
    /* Replace 1 with missing value to prevent error */
    CDF = choose(CDF > 1 - 2e-6, ., CDF);
    quant = quantile( "Normal", cdf );
    do j = 1 to ncol(P); /* set upper quantile to .I = infinity */
    quant[N[j],j] = .I; /* .I has special meaning to BIN func */
    end;
    return( quant );
finish;
/* OrdFindRoot: Use bisection to find the MV normal correlation that
produces a specified MV ordinal correlation. */
start OrdFindRoot(P1, P2, target);
    N1 = countn(P1); N2 = countn(P2);
    q1 = OrdQuant(P1); q2 = OrdQuant(P2);
    v1 = q1[1:N1-1]; v2 = q2[1:N2-1];
    g = Expand2DGrid(v1, v2);
```

```
        /* find value of rho so that sum(probbnrm(g[,1], g[,2], rho))=target */

        /* Bisection: find root on bracketing interval [a,b] */
        a = -1; b = 1; /* look for correlation in [-1,1] */
        dx = 1e-8; dy = 1e-5;
        do i = 1 to 100; /* iterate until convergence */
        c = (a+b)/2;
        Fc = sum( probbnrm(g[,1], g[,2], c) ) - target;
        if (abs(Fc) < dy) | (b-a)/2 < dx then
        return(c);
        Fa = sum( probbnrm(g[,1], g[,2], a) ) - target;
        if Fa#Fc > 0 then a = c;
        else b = c;
        end;
        return (.); /* no convergence */
finish;

/* OrdMVCorr: Compute a MVN correlation matrix from the PMF and
the target correlation matrix for the ordinal variables. */
start OrdMVCorr(P, Corr);
        d = ncol(P);
        N = OrdN(P);
        mean = OrdMean(P);
        var = OrdVar(P);
        cdf = OrdCDF(P);
        R = I(d);
        do i = 1 to d-1;
                sumCDFi = sum(cdf[1:N[i]-1, i]);
                do j = i+1 to d;
                        sumCDFj = sum(cdf[1:N[j]-1, j]);
                        hStar = Corr[i,j] * sqrt(var[i]*var[j]) + mean[i]*mean[j]
                        - N[i]*N[j] + N[i]*sumCDFj + N[j]*sumCDFi;
```

```
                R[i,j] = OrdFindRoot(P[,i], P[,j], hStar);
                R[j,i] = R[i,j];
            end;
        end;
        return(R);
finish;
/* RandMVOrdinal:
N Number of desired observations from MV ordinal distribution,
P Matrix of PMF for ordinal vars. The j_th col is the j_th PMF.
Use missing vals if some vars have fewer values than others.
Corr Desired correlation matrix for ordinal variables. Not every
matrix is a valid as the correlation of ordinal variables. */
start RandMVOrdinal(N, P, Corr);
        d = ncol(P);
        C = OrdMVCorr(P, Corr); /* 1. compute correlation matrix, C */
        mu = j(1, d, 0);
        X = RandNormal(N, mu, C); /* 2. simulate X ~ MVN(0,C) */
        N = OrdN(P);
        quant = OrdQuant(P); /* compute normal quantiles for PMFs */
do j = 1 to d; /* 3. convert to ordinal */
        X[,j] = bin(X[,j], .M // quant[1:N[j],j]);
end;
return(X);
finish;

        /*input data*/
P = {0.2 0.1 0.35 0.2,
0.2 0.2 0.35 0.3,
0.2 0.4 0.1 0.3,
0.2 0.2 0.1 0.1,
0.2 0.1 0.1 0.1};
```

```
		Corr1={1 0.8 0.8 0.75,
		0.8 1 0.9 0.8,
		0.8 0.9 1 0.8,
		0.75 0.8 0.8 1};

		Corr2={1 0 0 0, 0 1 0 0, 0 0 1 0, 0 0 0 1};
	call randseed(&seed);
	X = RandMVOrdinal(100, P, &Corr);

	*print the data set;
	create simulated from X[colname={"x1" "x2" "x3" "x4" "x5"}];
	append from X;
	close simulated;
quit;

*create simulated data set with Y;

	data YSimulated;
		set simulated;
		call streaminit(&g);
		y = floor(0.05*X1 + 0.15*X2 + 0.2*X3 + 0.6*X4 + rand('normal'));
		if y < 1 then y=1;
		if y > 5 then y=5;
	run;
%mend;

/************************************************************
The following block of code concerns with the evaluation
of key drivers using the ML Regression method. The
macro grabs a data set called YSimulated and solves for
the beta coefficients and then gets their importance values,
inputting the question with the maximum value
```

```
as the key driver into a row of the compiled table
OUTTABLE
**********************************************************/

%MACRO mlr(regressors,outtable);
PROC REG DATA=YSimulated OUTEST=EST NOPRINT;
        MODEL Y = &regressors / Rsquare;
QUIT;

PROC SQL NOPRINT;
        Select _RSQ_ into :rsquare from Est;

        Insert into Allests
        Select &regs3 from Est;
QUIT;

        *transpose table of estimates then sort and name key and 2nd driver;
      PROC TRANSPOSE DATA=Est OUT=Transpose;
           VAR &Regressors;
      RUN;

                PROC SORT DATA=Transpose;
                        BY DESCENDING COL1;
                RUN;

                DATA Transpose;
                        Set Transpose;
                                If _n_=1 then Kind="Key Driver";
                                Else if _n_=2 then Kind="2nd Driver";
                                Else Kind="Not Driver";
                RUN;
```

```
    PROC SQL;
                        INSERT INTO &OUTTABLE
      SELECT "MLR" as Method, X LABEL="Question", Prop AS Driver,Kind,&Rsquare
        FROM (SELECT _name_ AS X, ABS(col1)/SUM(ABS(col1)) AS Prop, Kind
          FROM Transpose)
          Where Kind="Key Driver" or Kind="2nd Driver";
    QUIT;

%MEND;

/*************************************************************
The following block of code concerns with the evaluation
of key drivers using the Shapley Regression method. The
macro grabs a data set called YSimulated and solves for
the Shapley values and then gets their importance values,
inputting the question with the maximum Shapley value
as the key driver into a row of the compiled table
OUTTABLE
*************************************************************/

%MACRO shapley(regressors,outtable);

PROC REG DATA=YSimulated OUTEST=EST NOPRINT;
        MODEL Y = &regressors / SELECTION=Rsquare START=1
STOP=%sysfunc(countw(&regressors));
QUIT;

%let m=%SYSFUNC(countw(&regressors));

PROC SQL NOPRINT;
        Select _RSQ_ into :rsquare from Est having _RSQ_=MAX(_RSQ_);
QUIT;
```

```
        *load the regressors into macro variables;
        %DO k = 1 %TO &m;
                %LET x&k = %SCAN(&regressors,&k);
        %END;

        *the proc reg step done earlier does all possible regression;
        *the step below takes the APR output and makes an additional
        variable regs that prints which variables are used in each model
        as a space-delimited list;
        DATA Est;
                Array x(&m) &regressors;
                Array vreg(&m) $ vreg1-vreg&m;
                SET Est;
                %Do j=1 %to &m;
                        If x(&j) ne . then vreg(&j)="X&j";
                        Else vreg(&j)="";
                %End;
                regs=catx(" ", of vreg1-vreg&m);
                KEEP regs _IN_  _RSQ_;
        RUN;
%DO k = 1 %TO %SYSFUNC(countw(&regressors));
        PROC SQL;
                CREATE TABLE btable&k (c_IN_ NUM,cregs1 CHAR(30), cregs2
CHAR(30), cRSQ NUM);
        QUIT;
%END;

        %DO i = 1 %TO %SYSFUNC(countw(&regressors));
                PROC SQL;
                        INSERT INTO btable&i
                        SELECT _IN_,regs,"" AS regs2,_RSQ_
```

```
                FROM EST
                WHERE _IN_=1 AND REGS="X&i";
        QUIT;
        %DO j = 2 %TO %SYSFUNC(countw(&regressors));
                PROC SQL;
                INSERT INTO btable&i
                        SELECT a._IN_, regs1,regs2,_RSQ_-RSQ2 AS RSQ
                        FROM (SELECT _IN_,_RSQ_, regs AS regs1, monotonic()
AS rows
                                FROM EST
                                WHERE REGS CONTAINS "X&i" AND _IN_=&j)
AS a
                                        FULL JOIN
                                (SELECT _IN_,_RSQ_ AS Rsq2, regs AS regs2,
monotonic() AS rows
                                FROM EST
                                WHERE REGS NOT CONTAINS "X&i" AND
_IN_=%EVAL(&j-1)) AS b
                                        ON a.rows=b.rows;
                QUIT;
        %END;
%END;


PROC SQL;
        CREATE TABLE compiledshapley(Question char(8),Shapley NUM);
QUIT;


%DO i = 1 %TO %SYSFUNC(countw(&regressors));
        PROC SQL;
        INSERT INTO compiledshapley
                SELECT "&&X&i" AS Q, avg(AveRSQ) AS SHAPLEY&i
```

```
                FROM
                        (SELECT c_IN_,avg(cRSQ) AS AveRSQ
                        FROM btable&i
                        GROUP BY c_IN_);
        QUIT;
%END;

PROC SORT DATA=CompiledShapley;
        By Descending Shapley;
RUN;

DATA CompiledShapley;
        Set CompiledShapley;
        If _n_=1 then Kind="Key Driver";
        Else if _n_=2 then Kind="2nd Driver";
        Else Kind="Not Driver";
RUN;

PROC SQL;
        Insert into &outtable
        SELECT "Shapley", Question, Shapley AS Driver,Kind,&Rsquare
        FROM compiledshapley
        Where Kind="Key Driver" or Kind="2nd Driver";

        Drop table compiledshapley;
        %do k=1 %to &m;
                Drop table btable&k;
        %end;
QUIT;
%MEND;

%macro corrdiff;
```

```
        PROC IML;
        Corr1={1 0.8 0.8 0.75,
                0.8 1 0.9 0.8,
                0.8 0.9 1 0.8,
                0.75 0.8 0.8 1};

        USE simulated;
        READ ALL VAR {x1 x2 x3 x4} INTO sim;
        CLOSE simulated;

        difference=abs(corr1-corr(sim));
        corrline=repeat(0,1,16);

        k=0;
        DO i=1 to 4;
                DO J=1 to 4;
                        k=k+1;
                        corrline[k]=difference[i,j];
                END;
        END;

        EDIT CORRDIFFERENCES;
        APPEND from corrline;
        CLOSE CORRDIFFERENCES;
QUIT;
%mend;


/*********************************************************
The following lines of code creates a compilation table
for the Shapley and MLR values called Drivers, simulates
200 data sets, each time doing MLR and Shapley on each,
```

```
and inputting the results into Drivers.
**********************************************************/
%macro complete(regressors,corr);

/*for timing purposes*/
%let begin = %sysfunc(TIME()) ;

%LET regs2=%sysfunc(tranwrd(&regressors,%str( ),%str( num,)));
%LET regs3=%sysfunc(tranwrd(&regressors,%str( ),%str(,)));

PROC SQL;
	Create table allests (&regs2 num);
	Create table DRIVERS (Method char(8), Question char(8), Driver num, Class
char(10),Rsquare num);
	Create table CORRDIFFERENCES (col1 num, col2 num, col3 num, col4 num, col5
num, col6 num, col7 num, col8 num, col9 num, col10 num, col11 num, col12 num, col13
num, col14 num, col15 num, col16 num);
QUIT;

%do g=1 %to 400;
	%ordsimulate(&g,&corr);
	%corrdiff;
	%mlr(&regressors,drivers);
	%shapley(&regressors,drivers);
%end;

	proc sort data=drivers;
		by method class;
	run;

	title "Drivers Found By Shapley and MLR";
	proc freq data=drivers;
		table question;
```

```
        by method class;
run;

title "Mean Absolute Error for Correlation between X1 X2";
proc sql;
        select avg(col2) from corrdifferences;
quit;

title "Importance Estimate Performance of the Key Drivers";
proc means data=drivers;
        var driver;
        by method;
        where question="x4";
run;

title "Importance vs Fit Performance of X4";
proc sgplot data=drivers;
        scatter x=rsquare y=driver;
        by method;
        where question="x4";
run;

title "Estimate Performances for MLR";
proc means data=allests MEAN STDDEV MIN MAX CLM;
        var &regressors;
run;

title;
*delete table;
proc sql;
        drop table simulated, corrdifferences;
quit;
```

```
/*for timing purposes*/
%put Entire code was executed %sysfunc(putn(%sysevalf(%sysfunc(TIME())-
&begin.),mmss.));
%mend;

*collinear, no omitted;
%complete(x1 x2 x3 x4, Corr1);

*collinear, omitted x3;
%complete(x1 x2 x4, Corr1);

*noncollinear, no omitted;
%complete(x1 x2 x3 x4, Corr2);

*noncollinear, omitted x3;
%complete(x1 x2 x4, Corr2);
```