# Logistic Regression to Identify Organizational Opportunities in Customer Surveys using R

**Author:** Ted Kwartler, AVP Innovation, Liberty Mutual Insurance

**Abstract:** Historically, identifying specific focal points for improving the customer experience using information from customer surveys has proven to be difficult. Often the large quantity of responses and the nature of the responses themselves often do not translate into specific areas for improvement that are impactful for satisfaction. However using logistic regression one can identify the impact to overall customer satisfaction of various survey responses. This practical article demonstrates a method to analytically understand customer responses and create a compelling visualization using the R programming language.

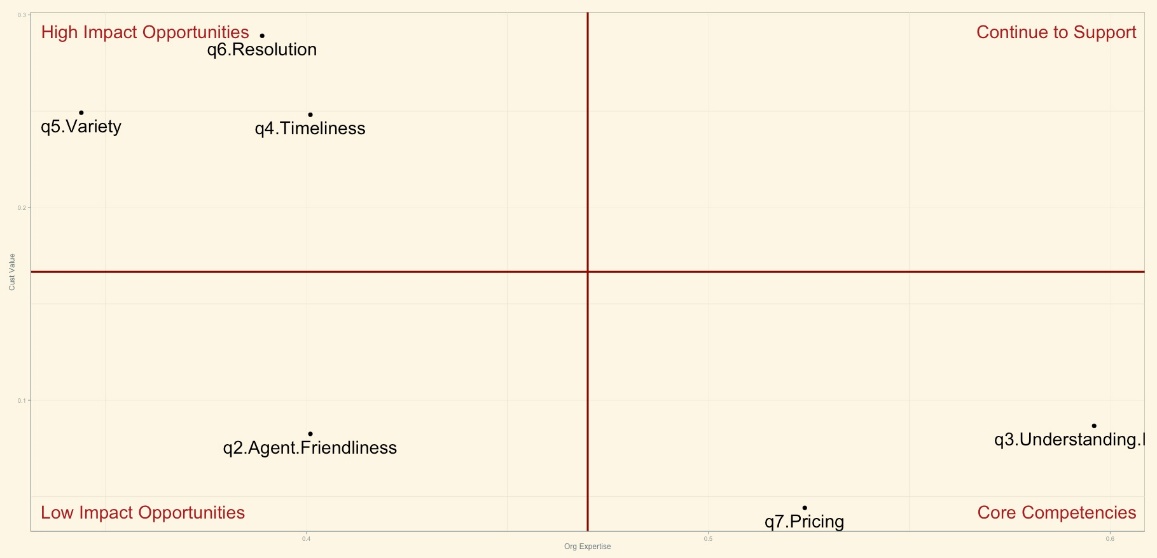
**Keywords:** Customer satisfaction, Customer survey, market research, logistic regression, R Programming

**Author Bio:** Ted Kwartler is a data-driven expert in analytics applied to production environments and text mining for actionable insights. While at amazon.com Ted used analytics for coaching, product launches, forecasting, and developing amazon’s social media customer care unit. Later he worked as a Director of Advanced Analytics focusing on a national customer service workforce for a Fortune 100 company. His team used analytics to improve efficiency, and understand the voice of the customer. Ted has worked in successful startups and also Fortune 100 companies in analytics roles to improve customer interactions. Ted holds an MBA from the University of Notre Dame with a citation in analytics and marketing.

## Background:

Forrester Research states some organizations are thriving in an age of customer obsession. In contrast, other organizations need to go through a transformation to change perspectives. In fact organizations must accept that “customers expect consistent and high-value in-person and digital experiences (1).” As a result, organizations that want to remain competitive must continually collect, analyze and create organizational insight from customer data. The challenge remains that there are often limited resources devoted to change and prioritizing aspects of a customer interaction can be elusive. One possible answer to this problem may be using logistic regression and examining the coefficients of customer surveys.

In practice articulating the findings of a linear model with many inputs can be problematic for decision makers. As a result the script in this practical article not only creates a logistic linear model but also a visualization to aid in understanding the results. Accurate and compelling visualizations allow even non-technical people to process the information and draw conclusions. By looking at each coefficient as a percent of the sum of all non-intercept coefficients one can understand the relative importance of that attribute from the customer perspective. In addition, reviewing the percent of all surveys in which the organization scored well in those attributes allows the organization to pinpoint aspects that are core competencies or opportunities. Figure 1 shows the end result of these two measurements in a compelling and concise visualization. In this applied article the explanations and example R code are provided.



*Figure 1 Example visual based on the fictitious survey data created from the script.*

## Practical Example:

Suppose you are an analyst for an organization in charge of extracting meaning from the multitude of customer satisfaction surveys your organization receives daily. The volume of the responses means that you could not read all responses and further your goal is to aid the operation to find areas for improvement thereby improving overall customer satisfaction.

For the sake of this example the customer survey consists of seven questions. Each question is a likert item where a customer choses a number between one and ten to measure the intended attribute. “A likert item is a statement that the respondent is asked to evaluate in a survey (2).” In this fictitious case the respondent is asked to select a number between one and ten with ten being the most desirable or best. Some market researchers prefer to use different scales or simply binary yes or no responses as is the case for amazon’s “did I resolve your issue?” In this illustrative example Table 1 represents an example of the questions in this fictional survey. The point of the exercise is not to debate scaling, or even questions themselves. Instead we will focus on extracting insights from survey results themselves and walk through the exact R code to do it.

### Example Survey Questions

|  |  |  |
| --- | --- | --- |
| Question ID | Question | Scale  (Low to High) |
| Q1 | Overall, how satisfied are you with the quality of service? | 1 - 10 |
| Q2 | How friendly was the agent you spoke with? | 1 - 10 |
| Q3 | How well did the agent understand your issue? | 1 - 10 |
| Q4 | How timely was the issue did we answer your initial inquiry? | 1 - 10 |
| Q5 | How satisfied are you with the product variety? | 1 - 10 |
| Q6 | How satisfied are you with the resolution of your inquiry? | 1 - 10 |
| Q7 | How satisfied are you with the pricing of the product you chose? | 1 - 10 |

*Table 1 Example survey questions used in the following example.*

In this case, a simple problem definition would state that a customers’ overall satisfaction with a service or product depends on the individual components of the measured components in the survey. Without bringing in computer telephony integrated information such as call timestamp, or wait time or using other contributing factors like customer longevity we are left with only the other measured information from the customer survey itself as feature inputs to the overall satisfaction. Admittedly this may pose a problem to the validity of the problem set up. These concerns are addressed in the criticism section of the article. However, the method outlined in this article is used in industry and can lead to operational insights despite the valid assessments discussed later. In essence this type of problem means that the dependent or y variable is represented by the first question covering overall satisfaction. The other questions contained in the survey represent the x or independent variables. And so it follows that:

)

A major assumption of this approach is the fact that while not truly independent attributes nonetheless the respondent is able to disassociate the measured attribute from the other ones on a question-by-question basis thereby making them independent. In truth a customer may have difficulty separating a friendly agent during a poor resolution and so on. As a result the method outlined here should be used in conjunction with other analyses.

## Data Preparation:

Given the questionnaire set up each observation would have six attributes and the dependent target. A logistic regression needs to have a binary outcome and so a judgment must be made to collapse the target into a one or zero. After examining the distribution and gaining some operational expertise it may make sense to collapse any Q1 Overall Satisfaction score that is a nine or ten as the success class represented by one. It then follows that all values in question one less than or equal to eight are considered a poor interaction. This step is handled in the code itself and is considered a feature-engineering step. Table two shows a brief example of the survey data.

### Representative Data from the example survey

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| SurveyID | Q1 Overall | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| 0001 | 8 | 3 | 7 | 2 | 3 | 7 | 9 |
| 0002 | 9 | 2 | 4 | 1 | 8 | 9 | 9 |
| 0003 | 9 | 3 | 6 | 10 | 3 | 9 | 9 |
| … | … | … | … | … | … | … | … |
| 1000 | 8 | 3 | 9 | 6 | 3 | 9 | 10 |

*Table 2 Truncated Example Survey data*

## Code explanation:

It is best to start all R code with some standardized version control information. This is because the practitioner may change roles or code later and having a way to organize files reduces the risk of lost information or recreating something from scratch. In this example I state the name, the reason for the code and information pertaining to the code author. The hash tag to begin lines means the code is not executable and merely comments to the overall script. Commenting using hash symbols is a basic functionality of R and used elsewhere in this script.

#Applied Marketing Analytics

#v1 Customer Satisfaction Survey & Logistic Regression

#Ted Kwartler; twitter: @tkwartler

Next, one should specify a working directory containing the survey data. In this example the code has a working directory explicitly set but this would need to be changed as you customize the code. The working directory is the path to the folder containing the survey data file. As a small tip one needs to take care to remember the slashes are revered in R compared to most operating systems and that the entire path is contained in quotes.

setwd('/Users/ted/Desktop/applied mktg')

After we build a model it is important to evaluate it. The next section of code creates a function to measure log loss. Log loss measures the inaccuracy of predicted probabilities to actuals based on the logistic regression output and known surveys. Later we also evaluate Area Under the Curve or AUC but to do so we do not need to create a custom function. We simply call the function later in the script.

#LogLoss Function

llfun <- function(actual, prediction) {

epsilon <- .000000000000001

yhat <- pmin(pmax(prediction, epsilon), 1-epsilon)

logloss <- -mean(actual\*log(yhat)

+ (1-actual)\*log(1 - yhat))

return(logloss)

}

The next custom function that is created will help create the x axis of the visual. This function is called “freqs”. In it, we pass in an object simply called x. In practice “freqs” is applied to the survey data questions two to seven. Step by step we manipulate the question data. First we change the integers one to ten to be treated as categorical factors. Then the function calls the summary function to sum each of the categories here sum of ones, twos and so on. In order to proceed we then change the object type to a matrix class. The next step adds the last row and the second to last row together. In this context it adds the number of respondents that gave the attribute a ten and also a nine. Lastly it takes that number and divides it by the total number of survey respondents. For the non R programmer this may be confusing but is explained further when the function is actually applied later.

freqs<-function(x){

y<-as.factor(x)

y<-summary(y)

y<-as.matrix(y)

y<-y[nrow(y),1]+y[(nrow(y)-1),1]

z<-y/length(x)

return(z)

}

The next section of code imports the appropriate R libraries to execute a logistic regression, evaluate output and then create a visualization.

#libraries

library(caret)

library(pROC)

library(LogicReg)

library(ggthemes)

Since this is a fabricated example we create survey responses artificially using the following code to create distributions. It is provided here to explicitly recreate the example in this article. In practice you would simple bring in a data frame similar to the one created in the “data” object using read.csv or something similar. The code first creates objects representing each individual question. Each of these objects is a random number distribution from one to ten with specific probabilities for each integer. The net result is that each individual question object is a numerical vector 1000 digits long. Lastly the individual objects are combined into a unified data frame simply called “data”.

#Example Data Creation

set.seed(1)

q1.Overall.Sat<-sample(1:10, 1000, replace=TRUE, prob=c(0.5,0.5,1,1,1,1,1,9.5,9.5,1))

q2.Agent.Friendliness<-sample(1:10, 1000, replace=TRUE, prob=c(10,9,10,1,1,1,1,1,9,10))#

q3.Understanding.Issues<-sample(1:10, 1000, replace=TRUE, prob=c(.5,.5,1,1,1,1,1,7,8.5,9))

q4.Timeliness<-sample(1:10, 1000, replace=TRUE, prob=c(5,1,.5,.5,.75,5,1,1,1,8))

q5.Variety<-sample(1:10, 1000, replace=TRUE, prob=c(1,1,10,1,1,1,.5,6,6,5))

q6.Resolution<-sample(1:10, 1000, replace=TRUE, prob=c(.25,.25,.25,1,10,10,10,1,10,10))#

q7.Pricing<-sample(1:10, 1000, replace=TRUE, prob=c(1,10,1,1,1,1,1,1,9,10))

data<-data.frame(q1.Overall.Sat,q2.Agent.Friendliness,q3.Understanding.Issues,

q4.Timeliness, q5.Variety,q6.Resolution,q7.Pricing)

Next we will need to engineer the dependent variable. As was mentioned earlier some understanding of question one’s distribution and what the organizations standards of a good interaction are needed to create a good cutoff. To make the new variable in the data object the script uses an ifelse statement. If the overall satisfaction number is nine or greater then create a one otherwise create a 0. The next line of code nullifies the question itself from the data frame. This ensures there is no target leakage when building a model based on this data frame.

#Target Creation

data$y<-ifelse(data$q1.Overall.Sat>=9, 1, 0)

data$q1.Overall.Sat<-NULL

Now that we have a modified data frame for modeling we must build the model itself. Using R’s ‘caret’ package by Max Kuhn the script executes a five fold cross validation. In a non synthetic example one should use a partition, validation and holdout set in addition to the nested cross validation shown here. After setting up the training control we train a model using caret and LogicReg packages. The syntax is train and then pass in the dependent variable simply called ‘y’. The next section in the train function must use a ‘~’ then pass all dependent variables to the function. Instead of listing each individually a simple period will tell the function to use all other columns in the specific data frame. Since caret can execute many types of models the script next identifies a generalized linear model using the ‘glm’ parameter. For a logistic regression we need to normalize the matrix. This is done next in the train function using the preprocess parameter. Specifically the parameters to perform normalization are “scale” and “center” applied to the modeling matrix. Next to train a model the data parameter is needed. It tells the function what training data to use. The trControl parameter affects the training method. Here it references the training control information located in the tc object. Lastly the train parameter needs to be told what type of linear model to construct. In this case the logit or logistic regression is applied.

tc <- trainControl("cv", 5, savePredictions=T)

fit<- train(y ~ .,

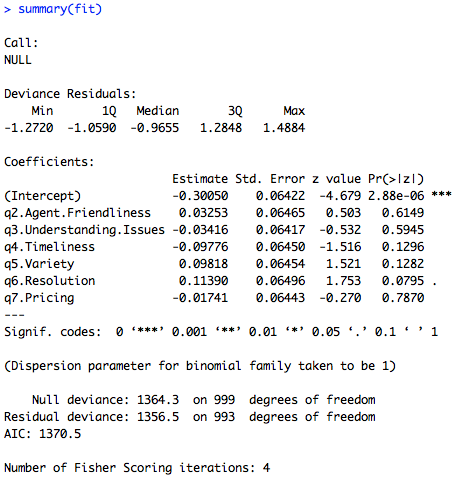
method='glm',

preProcess=c('scale', 'center'),

data=data, trControl = tc,

family=binomial(link='logit'))

The resulting ‘fit’ needs to be evaluated and this section of the script performs this action. As a reminder one should actually use a holdout set to test performance of the model. However since the data is not real this step was omitted. The summary function provides coefficients, p values and other important information of the fitted model.



*Figure 2 The summary of the fitted model object shows many useful diagnostics.*

In order to create a simple confusion matrix one needs to specify a cutoff probability so that probabilities can be classified as a one or zero. Here the code selects .42 as the cutoff. Always defaulting to .50 may not be best and care should be taken based on business case to balance accuracy, and cost of misclassification. The cutoff is an explicitly defined value and then used to classify the original data. This information is passed to a confusion matrix and then a simple arithmetic function helps calculate the overall model accuracy. Keep in mind since this accuracy represents non-holdout observations it is likely inflated. In machine learning this is considered prediction stacking and should be avoided in favor of holdout and validation folds. However as expected with randomly made inputs the model accuracy is really no better than a coin flip. Table 3 shows the confusion matrix output based on the code below.

cutoff<-.42 #change based on business case

pred.outcome<-ifelse(predict(fit,data)>=cutoff, 1, 0)

#Accuracy

confusionMatrix(pred.outcome,data$y)

sum(diag(table(pred.outcome,data$y)))/sum(table(pred.outcome,data$y))

### The confusion matrix of the fitted logistic regression

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Actual | |
|  |  | 0 | 1 |
| Prediction | 0 | 289 | 186 |
| 1 | 285 | 240 |

*Table 3 The confusion matrix based on the specific cut off probability.*

The script continues to evaluate the fitted object by invoking the customized log loss function made to begin the script. The log loss of this model is .678 and can be found by executing the R code below.

#LogLoss; lower is better, measures inaccuracy of predicted probabilities

llfun(data$y,predict(fit,data))

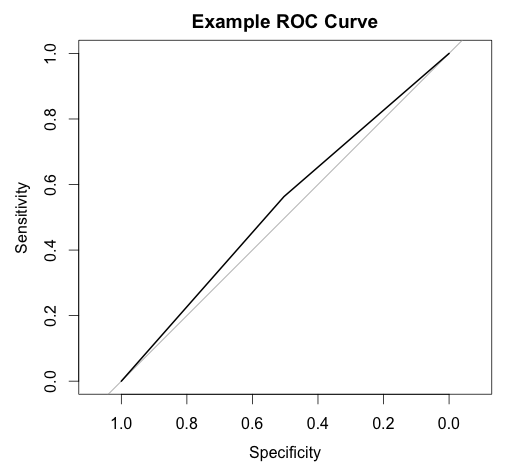
Continuing the model evaluation one could look at the area under the curve and the receiver operator curve itself. The pROC library conveniently wraps the computation for both into a simple function. This code will plot the ROC curve and return the AUC metric.

#AUC, ROC

roc.object <- roc(data$y, pred.outcome)

auc(roc.object)

plot(roc.object, main='Example ROC Curve')



*Figure 3 The simple ROC curve visual is not impressive but is expected given the random inputs that were generated.*

If one feels comfortable with the evaluation results using real data it then makes sense to create a visualization in order to derive a narrative of the data and help decision makers prioritize. To do so, we have to examine the model results in two dimensions. The first uses the logistic regression’s coefficients to understand customer value for the measurement. The second represents how often the organization performs well in that particular attribute.

First the coefficients are examined to determine customer valuation. The coefficients indicate the effects of the measurement on overall satisfaction. Thus a large and positive coefficient may indicate that a customer finds the particular attribute important. In contrast a small coefficient indicates that the impact to overall satisfaction is small. Since the sum of all betas represents the total possible amount of impact to overall satisfaction each attribute is individually compared to the coefficients sum. Taking an individual attribute coefficient and dividing by the sum of all coefficients will demonstrate the relative importance of the feature impact to overall satisfaction.

The next step is to examine how often an organization performs well in the measured attribute. To do so we revisit the custom “freqs” function. Using base R’s apply function on the data and using the freqs function we obtain a percent of all surveys that the organization scores a nine or ten for each question, column or attribute. This measurement helps to illustrate the questions that the organization is not scoring well consistently on within the survey data.

Both dimensions are combined in a small data frame called vis.data to be used in the visualizations. Table 4 shows the resulting data frame to be used for the illustration.

### The questions, their respective impact and how often the organization scores a 9 or 10

|  |  |  |
| --- | --- | --- |
| Question | Impact | Percent of All Interactions |
| q2.Agent.Friendliness | 0.08256583 | 0.401 |
| q3.Understanding.Issues | -0.08672153 | 0.596 |
| q4.Timeliness | -0.24816726 | 0.401 |
| q5.Variety | 0.24922906 | 0.344 |
| q6.Resolution | 0.28912335 | 0.389 |
| q7.Pricing | -0.04419296 | 0.524 |

*Table 4 is the data used in the visualization.*

#Basic Sum of Beta's Matrix

customer.impacts<-abs(fit$finalModel$coefficients[2:length(fit$finalModel$coefficients)]) /

sum(abs(fit$finalModel$coefficients[2:length(fit$finalModel$coefficients)]))

good.interactions<-apply(data[,1:6],2,freqs)

vis.data<-data.frame(questions=fit$coefnames,impact=customer.impacts, pct\_interactions=good.interactions)

The last step in this examination is to create a compelling visual using the vis.data information. The script identifies the mid point of both the x and y axis. This is so the scatter plot can be divided into quadrants for easier visual processing. In fact each quadrant is labeled as “Low Impact Opportunities”, “High Impact Opportunities”, ”Core Competencies” and “Continue to Support”. Low impact opportunities are those that are not highly valued by the customer thereby not supported by large coefficients and the organization does not score well in either. The high impact opportunity quadrant represents attributes with large coefficients meaning the customer values the attribute yet the organization does not perform well in consistently. Core competencies are low relative customer importance but the organization consistently performs. In fact questions appearing here may actually represent wasted effort. Lastly the fourth quadrant labeled “continue to support” represents attributes that the customer finds valuable and that the organization does well in. As a result the efforts here are valuable and should be maintained.

#Sum of Beta's 2 by 2

mid.x<-min(vis.data$pct\_interactions)+(max(vis.data$pct\_interactions)-min(vis.data$pct\_interactions))/2

mid.y<-min(vis.data$impact)+(max(vis.data$impact)-min(vis.data$impact))/2

annotations <- data.frame(

xpos = c(-Inf,-Inf,Inf,Inf),

ypos = c(-Inf, Inf,-Inf,Inf),

annotateText = c("Low Impact Opportunities","High Impact Opportunities","Core Competencies","Continue to Support"),

hjustvar = c(-.05,-.05,1.05,1.05), vjustvar = c(-1,2,-1,2))

ggplot(vis.data,aes(x=vis.data$pct\_interactions,y=vis.data$impact))+

geom\_text(aes(label=vis.data$questions), vjust=1.5,size=10)+

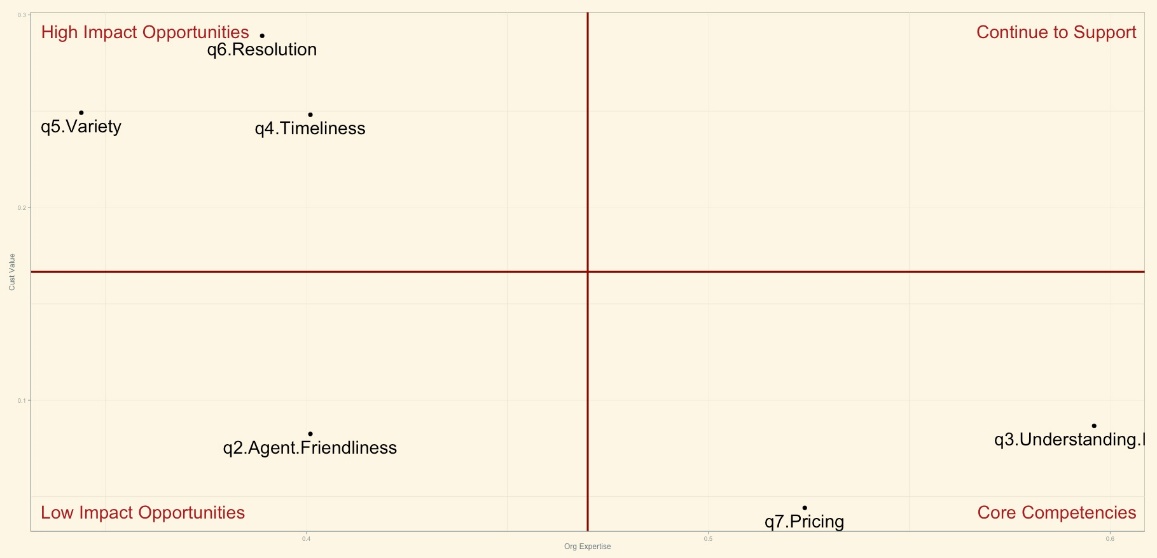
geom\_hline(aes(yintercept=mid.y), color='darkred', size=1.5)+

labs(x="Org Expertise", y="Cust Value", size=5)+

geom\_vline(aes(xintercept=mid.x), color='darkred', size=1.5)+theme\_solarized()+

theme(legend.position="none")+

geom\_text(data = annotations, aes(x=xpos,y=ypos,hjust=hjustvar,vjust=vjustvar, label=annotateText), color='#AC1D1C', size=10)+geom\_point(shape=19, size=3)



*Figure 4 demonstrates the visualization output with four labeled quadrants.*

Using the visual stakeholders can quickly understand that the customer does not value friendliness as much as issue resolution, timeliness of resolution and product or service variety. In contrast, the organization’s agent does a good job understanding the issue yet the customer does not value it. Of course the example here provides fake outputs and therefore do not make common sense. For example one may instead expect customers often care about pricing and friendliness.

## Criticism of this method:

There are drawbacks to using this method if you intend to base all operational improvement upon it. Specifically there are two shortcomings that are often cited with using the outlined approach with customer survey data. First there is a possibility for response bias among customer satisfaction surveys. Second the concept of simultaneity bias may also undermine the results.

Response bias can be defined as “conditions or factors that take place during the process of responding to surveys, affecting the way responses are provided” (3). This results in a non-random deviation of answers. If it is significant enough the customer survey data may not be an accurate portrayal of an organizational interaction and overall satisfaction with a product or service. In this case the response bias undermines the use of survey data for this and other analysis. For example human behavior often creates response bias in the form of leading questions on the part of the survey purveyor or social desirability on the part of the respondent. Assuming the questions of the survey are well thought out the aspect of social desirability still needs to be addressed. In practice social norms may mean people are unlikely to give negative scores thereby inflating the overall scores. This general niceness can negatively impact predictive accuracy. Further people with unusually good and bad interactions may be more likely to fill out a customer survey while others that received adequate service. Thus the extreme cases represent more “signal” in the data than is actually the case. Another form of response bias can occur if the respondent wants to reinforce their purchasing decision. Reinforcing customer behavior will again likely inflate the overall satisfaction with the good or product and have an impact on the model.

Simultaneity occurs because the respondent and collection of the attribute scores does not occur independently of each other (4). The debatable assumption is that a customer can disassociate question answers as they progress through the survey. Of course this is likely not completely the case and so the point is a fair one. However given an open dialogue and easy implementation of this method it makes sense to examine the surveys this way in conjunction with other market research methods.

As a result of these fair critiques caution should be taken to understand the human behaviors of the data and take steps to triangulate findings with other sources of customer information. In fact some researchers have modeled simultaneity within survey data using structural relationships in the data. Another solution may be to ensure the data analyst is consulted when drafting the response survey instead of having it solely in a marketing or customer service organization. In doing so, the analyst will be able to discuss shortcomings, expected methods for analysis and even help in wording and overall data integrity.

## Conclusion:

Despite the shortcomings in the methodology, using logistic regression in this manner may be useful and provide organizational insight. The effort to construct this visual is minimal and is about 100 lines of R code so it is worth exploring as part of a customer survey analysis. R is a popular language used in analytics but this method can be executed in other software environments including python, or SPSS. It is hoped that this practical example not only illustrates a way to understand survey results but also to present them to stakeholders in a compelling manner.

References:

1. Lead The Customer-Obsessed Transformation [Internet]. Forrester: In The Age of the Customer; 2015 [cited 2015 Dec 8]. Available from: https://www.forrester.com/age-of-the-customer/-/E-MPL291
2. Vanek Christian. Likert Scale – What Is It? When to Use It? How to Analyze It? [Internet]. SurveyGizmo; [updated 24 Apr. 2012; cited 2015 Dec 8] Availble from: <https://www.surveygizmo.com/survey-blog/likert-scale-what-is-it-how-to-analyze-it-and-when-to-use-it/>
3. Villar, Ana. Response Bias: Sage Research. Sage Research Methods. Sage Publications, 2008.
4. Gilbride, Timothy, Sha Yang, and Greg Allenby. Modeling Simultaneity in Survey Data. Quantitative Marketing and Economics 3.3 (2005): 311-35. Print.