


Article

A Generalized Linear Transformation and Its Effects on Logistic Regression

Guoping Zeng ^{1,*}  and Sha Tao ²¹ Independent Researcher, Plano, TX 75024, USA² School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, UK

* Correspondence: guopingtx@yahoo.com

Abstract: Linear transformations such as min–max normalization and z-score standardization are commonly used in logistic regression for the purpose of scaling. However, the work in the literature on linear transformations in logistic regression has two major limitations. First, most work focuses on improving the fit of the regression model. Second, the effects of transformations are rarely discussed. In this paper, we first generalized a linear transformation for a single variable to multiple variables by matrix multiplication. We then studied various effects of a generalized linear transformation in logistic regression. We showed that an invertible generalized linear transformation has no effects on predictions, multicollinearity, pseudo-complete separation and complete separation. We also showed that multiple linear transformations do not have effects on the variance inflation factor (VIF). Numeric examples with a real data were presented to validate our results. Our results of no effects justify the rationality of linear transformations in logistic regression.

Keywords: logistic regression; linear transformations; predictions; ordinary least squares estimator; maximum likelihood estimator

MSC: 62J12



Citation: Zeng, G.; Tao, S. A Generalized Linear Transformation and Its Effects on Logistic Regression. *Mathematics* **2023**, *11*, 467. <https://doi.org/10.3390/math11020467>

Academic Editor: Manuel Alberto M. Ferreira

Received: 20 November 2022

Revised: 8 January 2023

Accepted: 10 January 2023

Published: 15 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Logistic regression is one of the most commonly used techniques for modeling the relationship between the dependent variable and one or more independent variables.

In data analysis and machine learning, a transformation refers to a mapping of a variable into a new variable. A transformation can be linear or nonlinear, depending on whether the mapping is linear or nonlinear. Linear transformations can be used to improve interpretability of coefficients in linear regression and make a fitted model easier to understand [1], whereas nonlinear transformations are often used to improve the fit of the model on the data [2].

Three types of linear transformations are commonly used in machine learning prior to model fitting, namely, min–max normalization, z-score standardization and simple scaling. Since different variables that are measured in different scales may not contribute equally to model fitting, min–max normalization is used to transform all continuous variables into the same range [0, 1] to avoid a possible bias. Essentially, min–max normalization subtracts the minimum value of a continuous variable from each value and then divides by the range of the variable. z-score standardization rescales continuous variables to the standard scale, i.e., how far it is from the mean. Mathematically, z-score standardization subtracts the mean value of a continuous variable from each value and then divides by the standard deviation of the variable. Simple scaling shrinks or expands a continuous variable with big values and small values, respectively. The three types of linear transformations are all discussed by Adeyemo, Wimmer and Powell [3] for logistic regression.

However, the work in the literature on transformations in regression have some limitations. First, most work focuses on improving the fit of the regression model [4–9].

Second, the effects of transformations are rarely discussed. Morrell, Pearson, and Brant [10] examined how linear transformations affected a linear mixed-effect model and the tests of significance of fixed effects in the model. They showed how linear transformations modified the random effects, and their covariance matrix and the value of the restricted log-likelihood. Zeng [11] studied invariant properties of some statistical measures under monotonic transformations for univariate logistic regression. Zeng [12] derived analytic properties of some well-known category encodings such as ordinal encoding, order encoding and one-hot encoding in multivariate logistic regression by means of linear transformations. Adeyemo, Wimmer and Powell [3] compared the prediction accuracy of the three types of linear transformations, min–max normalization, z-score standardization and simple scaling, in logistic regression, by means of simulation.

In this paper, we first generalized a linear transformation for a single variable to multiple variables by a matrix multiplication. We then studied various effects of a generalized linear transformation in logistic regression. We showed that an invertible generalized linear transformation has no effects on predictions, multicollinearity, pseudo-complete separation, and complete separation. We also showed that multiple linear transformations do not have effects on the variance inflation factor (VIF). Numeric examples with randomly generated transformations from a real data were presented to illustrate our theoretic results.

The remainder of this paper is organized as follows. In Section 2, we give two definitions of a generalized linear transformation and show that they are equivalent. In Section 3, we study the effects of a generalized linear transformation on logistic regression. In Section 4, we present numeric examples to validate our theoretic results. Finally, the paper is concluded in Section 5.

Throughout the paper, we concentrate on transformations of independent variables, which are also sometimes called explanatory variables.

2. A Generalized Linear Transformation in Logistic Regression

Let $x = (x_1, x_2, \dots, x_p)$ be the vector of p independent variables and y be the dependent variable. Let us consider a sample of n independent observations $(x_{i1}, x_{i2}, \dots, x_{ip}, y_i)$, $i = 1, 2, \dots, n$, where y_i is the value of y and $x_{i1}, x_{i2}, \dots, x_{ip}$ the values of p independent variables x_1, x_2, \dots, x_p for the i -th observation. Without loss of generality, we assume x_1, x_2, \dots, x_p are all continuous variables since otherwise they can be converted into continuous variables.

Let us adopt the matrix notation:

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} x_{10} & x_{11} & \dots & x_{1p} \\ x_{20} & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n0} & x_{n1} & \dots & x_{np} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

where $x_{i0} = 1$ for all i (used for intercept β_0) and matrix X is called the design matrix. Here, $\beta_0, \beta_1, \dots, \beta_p$ are called regression coefficients or regression parameters.

Without causing confusion, we also use x_0, x_1, \dots, x_p to denote the $(p+1)$ columns or column vectors of X . We further use capital letter X_i to denote the row vector $(1, x_{i1}, x_{i2}, \dots, x_{ip})$ for $i = 1, 2, \dots, n$.

Definition 1. A linear transformation is a linear function of a variable which maps or transforms the variable into a new one. Specifically, a linear transformation of variable x can be defined as $x^t = ax + b$, where a and b are constants and a is nonzero. For convenience, let us call a linear transformation of a single variable a simple linear transformation. By multiple linear transformations, we mean a set of simple linear transformations. Here, we use letter t in the superscript to denote the new variable after a transformation.

Note that a and b in Definition 1 are not vectors since x is a variable.

Definition 1 can be generalized naturally by matrix multiplication to transform a set of variables to a new set of variables.

Definition 2. A generalized linear transformation is a linear matrix-vector expression

$$\begin{pmatrix} x_1^t \\ x_2^t \\ \vdots \\ x_p^t \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} \quad (1)$$

that transforms or maps independent variables x_1, \dots, x_p into new independent variables $x_1^t, x_2^t, \dots, x_p^t$, where $A = (a_{ij})$ is a $p \times p$ matrix of real numbers and b_1, b_2, \dots, b_p are real constants. Here, x_1, \dots, x_p are variables not vectors.

It should not be confused with the linear transformation between two vector spaces, in which there is no vector $b = (b_1, b_2, \dots, b_p)'$. Here and hereafter, we use the prime symbol ' in the superscript for the transpose of a vector or a matrix. The new variables $x_1^t, x_2^t, \dots, x_p^t$ in the component forms are

$$b_1 + \sum_{j=1}^p a_{1j}x_j, b_2 + \sum_{j=1}^p a_{2j}x_j, \dots, b_p + \sum_{j=1}^p a_{pj}x_j.$$

Consider a simple linear transformation, $x_i^t = ax_i + b$, for some x_i with $1 \leq i \leq p$. Without loss of generality, assume $i = 1$. Let A be a p -dimensional diagonal matrix with $a_{11} = a$ and $a_{22} = a_{33} = \dots = a_{pp} = 1$. Let $b = (b, 0, \dots, 0)'$ be a p -dimensional column vector. Then x_1, x_2, \dots, x_p are transformed into x_1^t, x_2, \dots, x_p according to Definition 2. Similarly, consider a set of simple linear transformations, say, $x_i^t = a_i x_i + b_i$ for $1 \leq i \leq r$ with $2 \leq r \leq p$. Let A be a p -dimensional diagonal matrix with $a_{ii} = a_i$ for $1 \leq i \leq r$ and $a_{ii} = 1$ for $i = r+1, r+2, \dots, p$. Let $b = (b_1, b_2, \dots, b_r, 0, \dots, 0)'$ be a p dimensional column vector. Then x_1, x_2, \dots, x_p are transformed into $x_1^t, x_2^t, \dots, x_r^t, x_{r+1}, x_{r+2}, \dots, x_p$ according to Definition 2. Hence, both a simple linear transformation and multiple linear transformations are a special case of a generalized linear transformation.

However, Definition 2 is not convenient to use since the new design matrix is somewhat complicated. Therefore, we give another definition incorporated with the design matrix.

Definition 3. A generalized linear transformation is a matrix multiplication XC that transforms x_1, \dots, x_p into $x_1^t, x_2^t, \dots, x_p^t$, where $x_1^t, x_2^t, \dots, x_p^t$ are the 2nd to the last column of XC and C is a $(p+1) \times (p+1)$ matrix of real numbers as follows

$$C = \begin{pmatrix} 1 & c_{11} & c_{12} & \dots & c_{1p} \\ 0 & c_{21} & c_{22} & \dots & c_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & c_{p1} & c_{p2} & \dots & c_{pp} \\ 0 & c_{p+1,1} & c_{p+1,2} & \dots & c_{p+1,p} \end{pmatrix}. \quad (2)$$

Note that we request the first column of C to be 0 except the first entry (which is 1) in order for XC to be the new design matrix.

For convenience, let us partition C into 4 blocks such that $C = \begin{pmatrix} 1 & C_1 \\ 0 & C_{11} \end{pmatrix}$, where C_1 is the p -dimensional row vector $(c_{11} \ c_{12} \ \dots \ c_{1p})$, 0 is the p -dimensional column vector of all 0 's and C_{11} is the $p \times p$ submatrix by deleting the first column and the first row of C , that is,

$$C_{11} = \begin{pmatrix} c_{21} & c_{22} & \dots & c_{2p} \\ c_{31} & c_{32} & \dots & c_{3p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p+1,1} & c_{p+1,2} & \dots & c_{p+1,p} \end{pmatrix}. \quad (3)$$

In the following we prove the definitions of generalized linear transformation are equivalent.

Theorem 1. *Definition 2 and Definition 3 are equivalent.*

Proof. Let us begin with Definition 2. Its new design matrix is

$$\begin{pmatrix} 1 & b_1 + \sum_{j=1}^p a_{1j}x_{1j} & b_2 + \sum_{j=1}^p a_{2j}x_{1j} & \dots & b_p + \sum_{j=1}^p a_{pj}x_{1j} \\ 1 & b_1 + \sum_{j=1}^p a_{1j}x_{2j} & b_2 + \sum_{j=1}^p a_{2j}x_{2j} & \dots & b_p + \sum_{j=1}^p a_{pj}x_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b_1 + \sum_{j=1}^p a_{1j}x_{nj} & b_2 + \sum_{j=1}^p a_{2j}x_{nj} & \dots & b_p + \sum_{j=1}^p a_{pj}x_{nj} \end{pmatrix} = X \begin{pmatrix} 1 & b_1 & b_2 & \dots & b_p \\ 0 & a_{11} & a_{21} & \dots & a_{p1} \\ 0 & a_{12} & a_{22} & \dots & a_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{1p} & a_{2p} & \dots & a_{pp} \end{pmatrix}. \quad (4)$$

Hence, the new design matrix of Definition 2 is in the form of Definition 3 with

$$C = \begin{pmatrix} 1 & b_1 & b_2 & \dots & b_p \\ 0 & a_{11} & a_{21} & \dots & a_{p1} \\ 0 & a_{12} & a_{22} & \dots & a_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{1p} & a_{2p} & \dots & a_{np} \end{pmatrix}. \quad (5)$$

Note that the submatrix by deleting the first row and first column of matrix C above is the transpose of A , that is, A' .

Next, let us begin with Definition 3.

$$\begin{aligned} XC &= X \begin{pmatrix} 1 & c_{11} & \dots & c_{1p} \\ 0 & c_{21} & \dots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & c_{p+1,1} & \dots & c_{p+1,p} \end{pmatrix} \\ &= \begin{pmatrix} 1 & c_{11} + \sum_{j=1}^p c_{j+1,1}x_{1j} & c_{12} + \sum_{j=1}^p c_{j+1,2}x_{1j} & \dots & c_{1p} + \sum_{j=1}^p c_{j+1,p}x_{1j} \\ 1 & c_{11} + \sum_{j=1}^p c_{j+1,1}x_{2j} & c_{12} + \sum_{j=1}^p c_{j+1,2}x_{2j} & \dots & c_{1p} + \sum_{j=1}^p c_{j+1,p}x_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & c_{11} + \sum_{j=1}^p c_{j+1,1}x_{nj} & c_{12} + \sum_{j=1}^p c_{j+1,2}x_{nj} & \dots & c_{1p} + \sum_{j=1}^p c_{j+1,p}x_{nj} \end{pmatrix}. \end{aligned} \quad (6)$$

The second, third, \dots , last column of the matrix above are from the linear transform

$$c_{11} + \sum_{j=1}^p c_{j+1,1}x_j, c_{12} + \sum_{j=1}^p c_{j+1,2}x_j, \dots, c_{1p} + \sum_{j=1}^p c_{j+1,p}x_j,$$

respectively. Hence, Definition 3 is in the form of Definition 2 with

$$A = \begin{pmatrix} c_{21} & c_{31} & \cdots & c_{p+1,1} \\ c_{22} & c_{32} & \cdots & c_{p+1,2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{2p} & c_{3p} & \cdots & c_{p+1,p} \end{pmatrix} \quad (7)$$

and

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} c_{11} \\ c_{12} \\ \vdots \\ c_{1p} \end{pmatrix}. \quad (8)$$

We have concluded our proof. \square

If we expand along the first column to find the determinant of C in (2), we immediately see that the determinant of C is equal to the determinant of C_{11} . Therefore, C is nonsingular (or invertible) if and only if C_{11} in (3) is nonsingular. In addition, it follows from the proof of Theorem 1 that C is nonsingular if and only if A in Definition 2 is nonsingular.

Moreover, it is easy to see that if C_{11} is nonsingular then the inverse of C can be written as

$$C^{-1} = \begin{pmatrix} 1 & -C_1 C_{11}^{-1} \\ 0 & C_{11}^{-1} \end{pmatrix}. \quad (9)$$

From now on we will use Definition 3 unless otherwise specified. For convenience, let us call the generalized linear transformation XC invertible if C is invertible.

3. Effects of a Generalized Linear Transformation

In logistic regression, the dependent variable y is binary with 2 values 0 and 1. Let the conditional probability that $y = 1$ be denoted by $\text{Prob}(y = 1 | x) = \pi(x)$.

Logistic regression assumes the logit linearity between the log odds and independent variables x_1, x_2, \dots, x_p

$$\ln \left[\frac{\pi(x)}{1 - \pi(x)} \right] = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p. \quad (10)$$

Equation (10) above can be written as

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}. \quad (11)$$

The following log likelihood is used in logistic regression

$$l(\beta, Y, X) = \sum_{i=1}^n y_i \ln \left(\frac{e^{X_i \beta}}{1 + e^{X_i \beta}} \right) + \sum_{i=1}^n (1 - y_i) \ln \left(\frac{1}{1 + e^{X_i \beta}} \right). \quad (12)$$

The maximum likelihood method is used to estimate parameters in logistic regression. Specifically, the maximum likelihood estimators (MLE) are the values of parameters $\beta_0, \beta_1, \dots, \beta_p$ that maximize (12). The vector $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p)'$ of the MLE estimators of $\beta = (\beta_0, \beta_1, \beta_2, \dots, \beta_p)'$ satisfies [13]

$$\begin{aligned} \sum_{i=1}^n y_i &= \sum_{i=1}^n \pi_i \\ \sum_{i=1}^n x_{ij} y_i &= \sum_{i=1}^n x_{ij} \pi_i \end{aligned} \quad (13)$$

or in matrix-vector form

$$X'Y = X'\pi \quad (14)$$

where $\pi = (\pi_1, \pi_2, \dots, \pi_n)'$, and $\pi_i = \pi(X_i) = \frac{e^{\hat{\beta}_i}}{1 + e^{\hat{\beta}_i}}$ for $i = 1, 2, \dots, n$. Note that after a generalized linear transformation XC , (12) and (14) hold with the design matrix X replaced by the new design matrix XC .

Equation (13) or (14) represents $(p + 1)$ nonlinear equations of $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ and cannot be solved explicitly in general [14]. Rather, they can be solved numerically by Newton-Raphson algorithm [15] as follows

$$\beta^{(i+1)} = \beta^{(i)} + (X'VX)^{-1}g, \quad (15)$$

where V is the $n \times n$ diagonal matrix with its diagonal elements $\pi_1(1 - \pi_1), \pi_2(1 - \pi_2), \dots, \pi_n(1 - \pi_n)$. In addition $g = X'(Y - \pi)$. Both V and g are evaluated at $\beta^{(i)}$ in (15).

If $X'X$ is nonsingular and the data is not completely separable or pseudo-completely separable [16], then the MLE estimator $\hat{\beta}$ exists and is unique.

The MLE estimator $\hat{\beta}$ can be used to predict $\text{prob}(y = 1|x)$ by the linear combination of variables x_1, \dots, x_p

$$\hat{\pi}(x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p}} = \frac{e^{(1, x_1, x_2, \dots, x_p)\hat{\beta}}}{1 + e^{(1, x_1, x_2, \dots, x_p)\hat{\beta}}}. \quad (16)$$

In particular, we have n fitted values

$$\hat{\pi}_i = \frac{e^{(1, x_{i1}, x_{i2}, \dots, x_{ip})\hat{\beta}}}{1 + e^{(1, x_{i1}, x_{i2}, \dots, x_{ip})\hat{\beta}}}, \quad i = 1, 2, \dots, n.$$

3.1. Effects on MLE Estimator and Predictions

Theorem 2. For logistic regression, if the MLE estimator of β is $\hat{\beta}$, then the MLE estimator of β is $C^{-1}\hat{\beta}$ after a generalized linear transformation XC assuming C is nonsingular. Moreover, the generalized linear transformation does not affect predictions.

Proof. Since $\hat{\beta}$ is the maximum likelihood estimator of β , (14) is satisfied by $\hat{\beta}$. Multiplying both sides of (14) by C' , we obtain

$$C'(X'Y) = C'(X'\pi). \quad (17)$$

Clearly, (17) can be rewritten as

$$(XC)'Y = (XC)'\pi. \quad (18)$$

Writing $X_i\hat{\beta}$ as $(X_iC)\left(C^{-1}\hat{\beta}\right)$ for $i = 1, 2, \dots, n$, we have

$$\pi_i = \frac{e^{X_i\hat{\beta}}}{1 + e^{X_i\hat{\beta}}} = \frac{e^{(X_iC)(C^{-1}\hat{\beta})}}{1 + e^{(X_iC)(C^{-1}\hat{\beta})}}. \quad (19)$$

It follows from (18) and (19) that $C^{-1}\hat{\beta}$ satisfies (14) for the new design matrix XC . Hence, the linear combinations $C^{-1}\hat{\beta}$ of $\hat{\beta}$ is the new MLE estimator after the generalized linear transformation XC .

Let us now predict $\text{prob}(y = 1)$ for a set of values of variables x_1, \dots, x_p , for the new system after the generalized linear transformation XC using the new MLE estimator $C^{-1}\hat{\beta}$. Let v_1, v_2, \dots, v_p be a specific value of x_1, \dots, x_p , respectively. Then, the row vector $(1, v_1, \dots, v_p)$ in the original system becomes $(1, v_1, \dots, v_p)C$ in the new system. By (16), the predicted conditional probability of $y = 1$ when $x = (1, v_1, \dots, v_p)C$ in the new system is

$$\frac{e^{(1, v_1, \dots, v_p)C(C^{-1}\hat{\beta})}}{1 + e^{(1, v_1, \dots, v_p)C(C^{-1}\hat{\beta})}} = \frac{e^{(1, v_1, \dots, v_p)\hat{\beta}}}{1 + e^{(1, v_1, \dots, v_p)\hat{\beta}}}. \quad (20)$$

The right-hand side of (20) is the predicted conditional probability of $y = 1$ when $x = (1, v_1, \dots, v_p)$ in the original system. \square

3.2. Effects on Multicollinearity

Perfect multicollinearity or complete multicollinearity or multicollinearity, in short, refers to a situation in logistic regression in which two or more independent variables are linearly related [17]. In particular, if two independent variables are linearly related, then it is called collinearity.

Mathematically, multicollinearity means there exist constant a_0, a_1, \dots, a_p such that

$$a_0x_0 + \sum_{i=1}^p a_ix_i = 0 \quad (21)$$

where at least two of a_1, \dots, a_p are nonzero. If we treat x_0 as an independent variable, then we just require at least one of a_1, \dots, a_p is nonzero.

Multicollinearity is a common issue in logistic regression. If there is multicollinearity, the design matrix X will not have a full column rank of $p + 1$. Hence, the $(p + 1) \times (p + 1)$ matrix $\hat{I} = X'VX$ in (15) will have a rank less than $p + 1$. Thus, the inverse matrix \hat{I}^{-1} in (15) does not exist, which make the iteration in (15) impossible.

If there is near multicollinearity and there is no separation of the data points, theoretically $\hat{I} = X'VX$ in (15) has an inverse and the iteration in (15) can be proceeded. Yet, iteration (15) may not find an approximate inverse $\hat{I} = X'VX$ and hence may cause unstable estimates and inaccurate variances [18].

Some authors define multicollinearity in logistic regression to be a high correlation between independent variables [19–21]. Let us call multicollinearity with high correlation by near multicollinearity and reserve multicollinearity for perfect multicollinearity or complete multicollinearity.

Let us define VIF now. Let R_j^2 be the R-squared that results when x_j is linearly regressed against the other $(p - 1)$ independent variables. Then VIF for x_j is defined as

$$VIF_j = \frac{1}{1 - R_j^2}, \quad j = 1, 2, \dots, p. \quad (22)$$

Near multicollinearity can be detected by using VIF [2]. The larger the VIF of an independent variable, the larger the correlation between this independent variable and others. However, there is no standard for acceptable levels of VIF. Multicollinearity can be combated by a generalized cross-validation (GCV) criterion in partially linear regression models [22,23].

3.2.1. Preliminary Results in Linear Regression

As VIF is related to linear regression, let us briefly introduce some preliminary results in linear regression. As for logistic regression, we consider p independent variables x_1, x_2, \dots, x_p . Unlike logistic regression, the dependent variable y in linear regression is a continuous variable. We shall adopt the same notation as in logistic regression unless otherwise specified. In particular, X is the design matrix.

In linear regression, the relationship between y and x_1, \dots, x_p is formulated as a linear combination

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon \quad (23)$$

where ϵ is a random error, or in matrix notation

$$Y = X\beta + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}. \quad (24)$$

The ordinary least squares (OLS) estimator $\hat{\beta}$ of β satisfies [2]

$$X'X\hat{\beta} = X'Y. \quad (25)$$

Assuming the $(p+1)$ -dimensional square matrix $X'X$ is nonsingular, then the OLS estimator $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p)'$ is unique and can be written explicitly as

$$\hat{\beta} = (X'X)^{-1}X'Y. \quad (26)$$

The OLS estimator $\hat{\beta}$ can be used to predict y by the linear combination of variables x_1, \dots, x_p as follows

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p = (1, x_1, x_2, \dots, x_p)\hat{\beta}. \quad (27)$$

Like Gelman and Hill [1] and Chatterjee and Hadi [2], we will call a predicted value a fitted value if the values of x_1, \dots, x_p come from one of the n observations. So, we have n fitted values

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \dots + \hat{\beta}_p x_{ip}, \quad i = 1, 2, \dots, n.$$

Therefore, the n -dimensional column vector \hat{Y} for the n fitted values $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ can be expressed as

$$\hat{Y} = \begin{pmatrix} \hat{y}_1 & \hat{y}_2 & \dots & \hat{y}_n \end{pmatrix}' = X\hat{\beta}. \quad (28)$$

It is easy to show that the OLS estimator is $C^{-1}\hat{\beta}$ after an invertible generalized linear transformation XC . Moreover, the generalized linear transformation does not affect predictions. Indeed, let us now predict y for a set of values of variables x_1, \dots, x_p , which could be from any set of values not necessarily from one of the n observations. We first transform the values of x_1, \dots, x_p into $(1, v_1, \dots, v_p)C$, where v_1, v_2, \dots, v_p are values of x_1, \dots, x_p . Next, we apply (27) and obtain $(1, v_1, \dots, v_p)C(C^{-1}\hat{\beta}) = (1, v_1, \dots, v_p)\hat{\beta}$, which is the predicted value of the original model.

In linear regression, the coefficient of determination, denoted by R^2 and also called R-squared, is given by Chatterjee and Hadi [2].

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (29)$$

where \bar{y} is the mean of the dependent variable y , that is, $\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$, and \hat{y}_i is the fitted value

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \dots + \hat{\beta}_p x_{ip}, \quad i = 1, 2, \dots, n.$$

The coefficient of determination R^2 can be related to the square of the correlation between Y and \hat{Y} as follows [2]

$$R^2 = \left[\text{Cor}(Y, \hat{Y}) \right]^2 \quad (30)$$

where

$$\text{Cor}(Y, \hat{Y}) = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}. \quad (31)$$

Theorem 3. R^2 in linear regression is invariant under invertible generalized linear transformations.

Proof. Expressing $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ in the numerator of the 2nd equation in (29) into the matrix form and applying (26), we obtain

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2 = (Y - X\hat{\beta})' (Y - X\hat{\beta}) = Y'Y - Y'X\hat{\beta}. \quad (32)$$

Substituting (32) into (29) yields

$$R^2 = 1 - \frac{Y'Y - Y'X\hat{\beta}}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (33)$$

Now let XC be an invertible generalized linear transformation. Then the OLS estimator after the transformation becomes $C^{-1}\hat{\beta}$. In this case, R^2 in (33) becomes

$$1 - \frac{Y'Y - Y'(XC)C^{-1}\hat{\beta}}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{Y'Y - Y'X\hat{\beta}}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

which returns to R^2 in (29) before the generalized linear transformation. \square

3.2.2. Effects on Logistic Regression

In Definitions 2 and 3, we defined a generalized linear transformation only for independent variables. Since an independent variable is used as the dependent variable in order to find its VIF, we consider a simple linear transformation for the dependent variable in the following result.

Since the determinant of D equals $a_1 a_2 \dots a_r \neq 0$, by Theorem 3, VIF for x_{r+1} after the generalized transformation ZD is the same as VIF for x_{r+1} prior to the generalized linear transformation. \square

Remark 1. VIFs are not necessarily invariant under an invertible generalized linear transformation XC . For instance, let $x_1^t = x_2$ and $x_2^t = x_1$ and keep x_3, x_4, \dots, x_p unchanged. Then $x_1^t, x_2^t, x_3, x_4, \dots, x_p$ result from the generalized linear transformation with

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Since the determinant of D is -1 , D is nonsingular. However, VIF for x_1^t after the generalizer linear transformation XD equals VIF for x_2 prior to the generalized linear transformation, which are unequal in general.

The following result is immediate.

Theorem 5. Multicollinearity exists in logistic regression if, and only if, it exists after an invertible generalized linear transformation.

Remark 2. All the results about multicollinearity and VIF also apply to machine learning algorithms in which multicollinearity is applicable such as linear regression.

3.3. Effects on Linear Separation

Albert and Anderson [16] first assumed design matrix X to have a full column rank, that is, no multicollinearity. They then introduced the concept of separation (including complete separation and quasi-complete separation) and overlap in logistic regression with intercept. They showed that separation leads to nonexistence of (finite) MLE and that overlap leads to finite and unique MLE. Therefore, like multi-collinearity, separation is a common issue in logistic regression.

Definition 4. There is a complete separation of data points if there exists a vector $b = (b_0, b_1, \dots, b_p)'$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \sum_{j=0}^p b_j x_{ij} = X_i b = b' X_i' > 0, & y_i = 1, \\ \sum_{j=0}^p b_j x_{ij} = X_i b = b' X_i' < 0, & y_i = 0. \end{cases} \quad (38)$$

Definition 5. There is quasi-complete separation if the data are not complete separable, but there exists a vector $b = (b_0, b_1, \dots, b_p)'$ such that

$$\begin{cases} \sum_{j=0}^p b_j x_{ij} = X_i b = b' X_i' \geq 0, & y_i = 1, \\ \sum_{j=0}^p b_j x_{ij} = X_i b = b' X_i' \leq 0, & y_i = 0 \end{cases} \quad (39)$$

and equality holds for at least one subject in each response group.

Definition 6. If neither a complete nor a quasi-complete separation exists, then the data is said to have overlap.

Theorem 6. An invertible generalized linear transformation does not affect the data configuration of logistic regression.

Proof. We consider three cases.

Case 1. There is a complete separation of data points in the original system. Then (38) holds for a vector $b = (b_0, b_1, \dots, b_p)'$. The row i in the design matrix is X_iC for $i = 1, 2, \dots, n$, after the invertible generalized linear transformation XC . Let $b^t = (C')^{-1}b$, then vector b^t is a constant column vector of dimension $(p + 1)$. Since $(b^t)'(X_iC)' = b'X_i'$, (38) holds after the generalized linear transformation. Therefore, there is also a complete separation of data points after the generalized linear transformation XC .

Case 2. There is a quasi-complete separation of data points in the original system. It can be proved similarly to Case 1.

Case 3. The original data point has overlap. Then the new data points after the generalized linear transformation XC also has overlap. We prove it by contradiction. Assume otherwise the new data points after the generalized linear transformation does not has overlap. Then there is either a complete separation or a pseudo-complete separation of data points. Let us first assume there is a complete separation of data point after the generalized linear transformation XC . Then there is a vector $b = (b_0, b_1, \dots, b_p)'$ such that (38) holds. Row i in the design matrix after the generalized linear transformation XC is X_iC for $i = 1, 2, \dots, n$. Let $b^t = (C')^{-1}b$, then (38) holds with b^t , which is a contradiction. Next, let us assume there is a quasi-complete separation after the generalized linear transformation XC . It can be proved similarly. \square

4. Numeric Examples

In this section, we use real data, the well-known German Credit Data from a German bank, to validate our theoretical results. The German Credit Data can be found in the UCI Machine Learning Repository [24]. The original dataset is in file “german.data”, which contains categorical/symbolic attributes. It has 1000 observations representing 1000 loan applicants. The statistical software package R (version 3.4.2) and its RStudio will be employed for our analyses. Since there are only 1000 records, we will not split them into training and test. We extract german.data using R’s read_table function, call it german_credit_raw, and use colnames() method to rename the column names.

There are 21 variables or attributes in german_credit_raw including 8 numerical ones as follows, which are denoted by x_1, x_2, \dots, x_8 , respectively:

- Duration: Duration in month;
- credit_amount: Credit amount;
- installment_rate: Installment rate in percentage of disposable income;
- current_address_length: Present residence since;
- age: Age in years;
- num_credits: Number of existing credits at this bank;
- num_dependents: Number of people being liable to provide maintenance for;
- credit_status: Credit status: 1 for good loans and 2 for bad loans.

Let us define a new variable called default as $y = \text{default} = \text{credit_status} - 1$. With the new variable default, 0 is for good loans and 1 is for bad loans. Since it is not easy to interpret categorical variables, we will only consider numerical variables.

4.1. Validation of Invariance of Separation

Let us first build a logistic regression model `logit_model_1` using all the 8 numerical variables and `glm` function in R. In the following, we italicize statements in R, use “>” for the R prompt and make outputs from R bold.

```
> logit_model_1 <- glm(default ~ duration + credit_amount + installment_rate +
  current_address_length + age + num_credits + num_dependents + credit_status, data =
  german_credit_raw, family = "binomial")
```

Warning message:

```
glm.fit: algorithm did not converge
```

We see a warning message as above. It indicates a separation in the data. Indeed, this separation is from variable `credit_status`. (38) holds with $b_0 = -3$, $b_1 = b_2 = \dots = b_7 = 0$, $b_8 = 2$. By Definition 4, there is a complete separation of data points.

Now let us make a generalized linear transformation. We randomly generate 8×8 matrix C_{11} as shown in Table 1 and the 8-dimensional row vector C_1 in (3) by calling R function `runif`, which generates random values from a uniform distribution with a default value from 0 to 1. We set seed for the purpose of reproduction. We denote C_{11} and C_1 by `C_11` and `C_1` in R, respectively. We call R’s function `det` to calculate the determinant of C_{11} .

Table 1. Matrix C_{11} .

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0.2655087	0.6291140	0.7176185	0.2672207	0.4935413	0.8209463	0.7323137	0.3162717
[2,]	0.3721239	0.0617863	0.9919061	0.3861141	0.1862176	0.6470602	0.6927316	0.5186343
[3,]	0.5728534	0.2059746	0.3800352	0.0133903	0.8273733	0.7829328	0.4776196	0.6620051
[4,]	0.9082078	0.1765568	0.7774452	0.3823880	0.6684667	0.5530363	0.8612095	0.4068302
[5,]	0.2016819	0.6870228	0.9347052	0.8696908	0.7942399	0.5297196	0.4380971	0.9128759
[6,]	0.8983897	0.3841037	0.2121425	0.3403490	0.1079436	0.7893562	0.2447973	0.2936034
[7,]	0.9446753	0.7698414	0.6516738	0.4820801	0.7237109	0.0233312	0.0706790	0.4590657
[8,]	0.6607978	0.4976992	0.1255551	0.5995658	0.4112744	0.4772301	0.0994662	0.3323947

```
> set.seed(1)
```

```
> C_11 <- matrix(runif(64), nrow = 8)
```

We use R function `det` to find the determinant of C_{11} to be 0.01433565.

Vector C_1 is generated as follows:

```
> set.seed(10)
```

```
> C_1 = runif(n = 8, min = 1, max = 20)
```

```
[1] 10.642086 6.828602 9.111246 14.168940 2.617583 5.283296
```

```
6.216080 6.173796
```

Since C_{11} is nonsingular, so is $C = \begin{pmatrix} 1 & C_1 \\ 0 & C_{11} \end{pmatrix}$ by (9). Now x_1, \dots, x_8 can be transmitted into $x_1^t, x_2^t, \dots, x_8^t$ as in (6). Let us denote $x_1^t, x_2^t, \dots, x_8^t$ by *duration₂*, *credit_amount₂*, *credit_status₂* in R.

Let us build a logistic regression model `logit_model_2` for the eight transformed variables.

We also see the warning message as for the eight original variables. Therefore, after a nonsingular generalized linear transformation, the separation in data remains.

4.2. Validation of MLE

Let us drop `credit_status` and rebuild a logistic regression model called `logit_model_3`. The main output is shown in Table 2.

Table 2. Coefficients and statistics for model 3.

Coefficients:					
	Estimate	Std. Error	z Value	Pr(> z)	
(Intercept)	−1.56979765	0.42997660	−3.651	0.000261	***
duration	0.02621174	0.00770330	3.403	0.000667	***
credit_amount	0.00007060	0.00003404	2.074	0.038053	*
installment_rate	0.20355992	0.07251671	2.807	0.004999	**
current_address_length	0.04090933	0.06690897	0.611	0.540923	
age	−0.02143075	0.00708337	−3.026	0.002482	**
num_credits	−0.15689020	0.13049965	−1.202	0.229276	
num_dependents	0.12800328	0.20131338	0.636	0.52488	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.

The output also indicates the data still has *overlap* after the transformation. Hence, we have validated Theorem 6.

We see variables `current_address_length`, `num_credits` and `num_dependents` are not significant at the 0.05 level. Since we are not focused on building a model, let us still keep these variables. Let us extract the coefficients and put them in a vector called `model_coef_3` as follows:

```
> model_coef_3 <- data.frame(coef(logit_model_3))
> model_coef_3 <- as.matrix(model_coef_3)
```

Next, let us make a generalized linear transformation. We use letter D rather than C to distinguish the case from Section 4.1. We randomly generate 7×7 matrix D_{11} and the 7-dimensional row vector D_1 in (3) by calling R function `runif`. Again, we denote D_{11} as shown in Table 3 and D_1 by `D_11` and `D_1` in R, respectively.

Table 3. Matrix D_{11} .

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.1848823	0.8334488	0.40528218	0.3875495	0.96264405	0.6271963	0.1150841
[2,]	0.7023740	0.4680185	0.85354845	0.8368892	0.13237200	0.8444290	0.1632009
[3,]	0.5733263	0.5499837	0.97639849	0.1505014	0.01041453	0.2848706	0.9440418
[4,]	0.1680519	0.5526741	0.22582546	0.3472722	0.16464224	0.6672256	0.7948638
[5,]	0.9438393	0.2388948	0.44480923	0.4887732	0.81019214	0.1504698	0.9746879
[6,]	0.9434750	0.7605133	0.07497942	0.1492469	0.86886104	0.9817279	0.3490884
[7,]	0.1291590	0.1808201	0.66189876	0.3570626	0.51428176	0.2970107	0.5019699

```
> set.seed(2)
> D_11 <- matrix(runif(49),nrow = 7)
> det(D_11)
[1] 0.2851758
> set.seed(20)
> D_1 = runif(n = 7, min = 1, max = 20)
[1] 17.672906 15.602131 6.300300 11.054110 19.295234 19.626737
2.735319
```

Since the determinant of D_{11} is nonzero, $D = \begin{pmatrix} 1 & D_1 \\ 0 & D_{11} \end{pmatrix}$ is non-singular by (9) as shown in Table 4:

Table 4. Matrix D.

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1	17.6729064	15.6021310	6.30029963	11.0541102	19.29523359	19.6267371	2.7353192
[2,]	0	0.1848823	0.8334488	0.40528218	0.3875495	0.96264405	0.6271963	0.1150841
[3,]	0	0.7023740	0.4680185	0.85354845	0.8368892	0.13237200	0.8444290	0.1632009
[4,]	0	0.5733263	0.5499837	0.97639849	0.1505014	0.01041453	0.2848706	0.9440418
[5,]	0	0.1680519	0.5526741	0.22582546	0.3472722	0.16464224	0.6672256	0.7948638
[6,]	0	0.9438393	0.2388948	0.44480923	0.4887732	0.81019214	0.1504698	0.9746879
[7,]	0	0.9434750	0.7605133	0.07497942	0.1492469	0.86886104	0.9817279	0.3490884
[8,]	0	0.1291590	0.1808201	0.66189876	0.3570626	0.51428176	0.2970107	0.5019699

We use R function solve to find its inverse D^{-1} and call it inv_D (see Table 5) in R

Table 5. Inverse matrix of D.

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	1	−7.9987564	−8.13990103	4.21471891	1.9118435	−3.7229455	−10.65351722	2.7150084
[2,]	0	−0.3069549	0.33838941	0.27554240	−0.6317051	0.4987514	0.42693364	−0.8228946
[3,]	0	1.6396322	−0.06252947	0.79806766	0.2807233	0.2680550	−0.76439514	−2.2899087
[4,]	0	−0.1062208	0.10961077	0.64555254	−0.8488122	−0.5625429	0.10509829	1.1379418
[5,]	0	0.6460218	0.86107501	−0.79420393	0.7167836	0.9646552	−1.22876225	−1.0880130
[6,]	0	0.2595927	−0.43338042	−0.38521513	−0.4283198	0.1580233	0.29055578	0.9751883
[7,]	0	−1.1460418	0.12889195	−0.48351108	0.4179223	−1.0293070	1.14940313	1.6676878
[8,]	0	−0.4189740	−0.45383398	0.03608131	0.8623441	0.2778314	−0.07680999	0.3163327

Now x_1, \dots, x_7 can be transmitted into $x_1^t, x_2^t, \dots, x_7^t$ as in (6). Let us denote $x_1^t, x_2^t, \dots, x_7^t$ by *duration*₄, *credit_amount*₄, ..., *number_dependents*₄ in R. Let us build a logistic regression model for the seven transformed variables and call it *logit_model_4*. The main output is shown in Table 6:

Table 6. Coefficients and statistics of model 4.

Coefficients:	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.25487	1.65128	0.760	0.447
duration_4	−0.16078	0.18457	−0.871	0.384
credit_amount_4	0.03798	0.46388	0.082	0.935
installment_rate_4	0.23513	0.24465	0.961	0.337
current_address_length_4	−0.08251	0.27582	−0.299	0.765
age_4	−0.01331	0.19983	−0.067	0.947
num_credits_4	−0.05616	0.35614	−0.158	0.875
num_dependents_4	0.07820	0.08644	0.905	0.366

Let us extract the coefficients called *model_coef_3* to get more digits as shown in Table 7:

Table 7. Coefficients of model 4.

	coef.logit_model_4.
(Intercept)	1.2548745
duration_4	−0.1607787
credit_amount_4	0.0379776
installment_rate_4	0.2351349
current_address_length_4	−0.0825126
age_4	−0.0133075
num_credits_4	−0.0561589
num_dependents_4	0.0781968


```
> model_coef_4 <- data.frame(coef(logit_model_4))
```

Let us find the multiplication of D^{-1} and vector model_coef_3 in R as follows:

```
> inv_D%%model_coef_3
```

The result of the product is shown in Table 8 below.

Table 8. Product of inverse matrix of D and coefficients of model 3.

	coef.logit_model_3.
[1,]	1.25487449
[2,]	−0.16077871
[3,]	0.03797764
[4,]	0.23513490
[5,]	−0.08251257
[6,]	−0.01330746
[7,]	−0.05615895
[8,]	0.07819677

This is exactly the same as model_coef_4. Next, we calculate the predicted values for all the 1000 records using both models logit_model_3 and logit_model_4 by calling R function predict and then all.equal utility to check these two predictions are near equality:

```
> model_3_predictions = predict(logit_model_3, german_credit_raw, type="response")
```

```
> model_4_predictions = predict(logit_model_4, german_credit_raw, type="response")
```

```
> all.equal(model_3_predictions, model_4_predictions, tolerance = 1e-13)
```

```
[1] "Mean relative difference: 0.0000000000005060054"
```

We see that the two predictions are identical taking rounding errors into consideration. Thus, we have validated Theorem 2.

Note that a nonlinear transformation even a one-to-one correspondence will not have the properties in Theorem 2 even for a single variable. For instance, let us define a one-to-one correspondence for variable age as follows: $age_6 = \ln(age)$, which is $\log(age)$ in R. Let us build a univariate logistic regression model called logit_model_5 for age and a univariate logistic regression model called logit_model_6 for age_6. Next, we apply these two models to predict the values for german_credit_raw.

```
> model_5_predictions = predict(logit_model_5, german_credit_raw, type="response")
```

```
> model_6_predictions = predict(logit_model_6, german_credit_raw, type="response")
```

```
> all.equal(model_5_predictions, model_6_predictions, scale=1)
```

```
[1] "Mean absolute difference: 0.008512868"
```

We see that the predictions from logit_model_5 are in general different from predictions for logit_model_6.

4.3. Validation of Invariance of VIF

For logistic regression model logit_model_3 in Section 4.2, we use VIF function in the car package of R to find VIF for all the 7 variables. The result is shown in Table 9.

Table 9. VIF for model 3.

	VIF
duration	1.781992
credit_amount	1.991715
installment rate	1.223520
curent_adress_length	1.069032
age	1.111486
num_credits	1.031390
num_dependents	1.033490

```
> car::vif(logit_model_3)
```

Next, we randomly generate multiple simple transformations as follows

```

> set.seed(30)
> A = runif(n = 7)
> set.seed(40)
> B = runif(n = 7, min = 1, max = 10)
> german_credit_raw$duration_7 = A [1] * german_credit_raw$duration + B [1]
> german_credit_raw$credit_amount_7 = A [2] * german_credit_raw$credit_amount +
B [2]
...
> german_credit_raw$num_dependents_7 = A [7] * german_credit_raw$num_dependents
+ B [7]

```

We build a logistic regression for the variables after multiple simple linear transformations and call it `logit_model_7`. We then find VIF as follows and display the result in Table 10

Table 10. VIF for model 7.

	VIF
duration_7	1.781992
credit_amount_7	1.991715
installment_rate_7	1.223520
curent_adress_length_7	1.069032
age_7	1.111486
num_credits_7	1.031390
num_dependents_7	1.033490

```
> car::vif(logit_model_7)
```

Hence, we have validated Theorem 4. There is no need to validate Theorem 5 (the invariance of multicollinearity) as its analytical proof is straightforward.

5. Conclusions

In this paper, we first generalized a linear transformation for a single variable to multiple variables by a matrix multiplication. We then studied various effects of a generalized linear transformation in logistic regression. We showed that an invertible generalized linear transformation has no effects on predictions, multicollinearity, pseudo-complete separation, and complete separation. We also showed that multiple linear transformations do not have effects on the variance inflation factor (VIF). Numeric examples with real data were presented to validate our theoretic results.

Author Contributions: Writing—original draft, G.Z.; Writing—review & editing, S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: <http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/>, accessed on 6 November 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gelman, A.; Hill, J. *Data Analysis Using Regression and multilevel/Hierarchical Models*; Cambridge University Press: Cambridge, UK, 2006.
2. Chatterjee, S.; Hadi, A.S. *Regression Analysis by Example*, 5th ed.; John Wiley & Sons: New York, NY, USA, 2013.
3. Adeyemo, A.; Wimmer, H.; Powell, L.M. Effects of normalization techniques on logistic regression in data science. *J. Inf. Syst. Appl. Res.* **2019**, *12*, 37–44.
4. Box, G.E.P.; Tidwell, P.W. Transformation of the Independent Variables. *Technometrics* **1962**, *4*, 531–550. [[CrossRef](#)]

5. Whittemore, A.S. Transformations to Linearity in Binary Regression. *SIAM J. Appl. Math.* **1983**, *43*, 703–710. [[CrossRef](#)]
6. Kay, R.; Little, S. Transformations of the explanatory variables in the logistic regression model for binary data. *Biometrika* **1987**, *74*, 495–501. [[CrossRef](#)]
7. Feng, C.; Wang, H.; Lu, N.; Chen, T.; He, H.; Lu, Y.; Tu, X.M. Log-transformation and its implications for data analysis. *Shanghai Arch. Psychiatry* **2014**, *26*, 105–109. [[PubMed](#)]
8. Zhang, M.; Chen, S.; Rain, S.C. Evaluating Continuous Variable Transformations in Logistic Regression. In Proceedings of the Midwest SAS User Group Conference 2015, Omaha, NE, USA, 18–20 October 2015.
9. Lee, D.K. Data transformation: A focus on the interpretation. *Korean J. Anesthesiol.* **2020**, *73*, 503–508. [[CrossRef](#)] [[PubMed](#)]
10. Morrell, C.H.; Pearson, J.D.; Brant, L.J. Linear Transformations of Linear Mixed-Effects Models. *Am. Stat.* **1997**, *51*, 338–343.
11. Zeng, G. Invariant Properties of Logistic Regression Model in Credit Scoring under Monotonic Transformations. *Commun. Stat. Theory Methods* **2017**, *46*, 8791–8807. [[CrossRef](#)]
12. Zeng, G. On the analytical properties of category encodings in logistic regression. *Commun. Stat. Theory Methods*, 2021; advance online publication. [[CrossRef](#)]
13. Hosmer, D.W.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*, 3rd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 2013.
14. Zeng, G. On the Existence of an Analytical Solution in Multiple Logistic Regression. *Int. J. Appl. Math. Stat.* **2021**, *60*, 53–67.
15. Refaat, M. *Credit Risk Scorecards: Development and Implementation Using SAS*; Lulu.com: Raleigh, NC, USA, 2011.
16. Albert, A.; Anderson, J.A. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **1984**, *71*, 1–10. [[CrossRef](#)]
17. Zeng, G.; Zeng, E. On the Relationship between Multicollinearity and Separation in Logistic Regression. *Commun. Stat. Simul. Comput.* **2021**, *50*, 1989–1997. [[CrossRef](#)]
18. Shen, L.; Gao, Y.; Xiao, J. Simulation of Hydrogen Production from Biomass Gasification in Interconnected Fluidized Beds. *Biomass Bioenergy* **2008**, *32*, 120–127. [[CrossRef](#)]
19. Vatcheva, K.P.; Lee, M.; McCormick, J.B.; Rahbar, M.H. Multicollinearity in Regression Analyses Conducted in Epidemiologic Studies. *Epidemiology* **2016**, *6*, 227. [[CrossRef](#)] [[PubMed](#)]
20. Cincotta, K. Multicollinearity in Zero Intercept Regression: They Are Not Who We Thought They Were. In Proceedings of the Presented at the Society of Cost Estimating and Analysis (SCEA) Conference, Albuquerque, NM, USA, 6–10 June 2011.
21. Dohoo, I.R.; Ducrot, C.; Fourichon, C.; Donald, A.; Hurnik, D. An overview of techniques for dealing with large numbers of independent variables in epidemiologic studies. *Prev. Vet. Med.* **1997**, *29*, 221–239. [[CrossRef](#)] [[PubMed](#)]
22. Amini, M.; Roozbeh, M. Optimal partial ridge estimation in restricted semiparametric regression models. *J. Multivar. Anal.* **2015**, *136*, 26–40. [[CrossRef](#)]
23. Roozbeh, M. Optimal QR-based estimation in partially linear regression models with correlated errors using GCV criterion. *Comput. Stat. Data Anal.* **2018**, *117*, 45–61. [[CrossRef](#)]
24. Lichman, M. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2013; Available online: <http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/> (accessed on 16 December 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.