Understanding and Resolving Singularity Errors in R Statistical Models

Authored by

Mohammed looti

November 2, 2025

One of the most challenging and fundamentally important error messages encountered during statistical modeling in R signals a critical structural flaw known as rank deficiency. When fitting a Generalized Linear Model (GLM), analysts may receive a concise but alarming warning that directly impacts the validity of the results:

Coefficients: (1 not defined because of singularities)

This warning is more than a simple nuisance; it indicates that the parameters of your model cannot be uniquely estimated due to redundancy among the predictor variables. The presence of singularities in the underlying mathematical structure makes coefficient calculation impossible. Resolving this issue requires a clear understanding of the statistical concept--perfect multicollinearity--and a systematic diagnostic approach.

The error typically surfaces when using the primary modeling function, the **glm()** function, and happens because two or more predictor variables share an exact linear relationship. When this perfect correlation exists, the crucial **design matrix** used for estimation becomes mathematically singular (non-invertible). Since the modeling algorithm relies on inverting this matrix to solve for the coefficients, the process fails, leading to undefined parameters.

Fortunately, identifying and rectifying this structural defect is straightforward. The solution involves using tools like the **cor()** function to pinpoint the redundant variables and subsequently removing one from the regression equation. This comprehensive guide provides a practical, step-by-step methodology for diagnosing the root cause of singularities and ensuring a valid model fit in R.

Diagnosing the Rank Deficiency Warning

The warning "not defined because of singularities" is R's technical way of stating that the **design matrix** (X) generated from your predictor variables lacks full rank. Full rank is a necessary condition for successful regression estimation, as it ensures that all columns (variables) are linearly independent. When a matrix is rank-deficient, it means at least one variable is a perfect linear combination of the others, rendering that variable statistically superfluous.

Statisticians refer to this exact duplication of information as perfect **multicollinearity**. Unlike high but imperfect correlation, which merely increases the variance of estimates (leading to large standard errors), perfect collinearity makes the matrix inversion required for estimation fundamentally impossible. When this occurs, R cannot solve the system of equations that define the coefficients, resulting in the coefficients for the redundant terms being reported as **NA** (Not Available).

It is vital to recognize that the number indicated in the warning--for example, "(1 not defined

because of singularities)"--tells you exactly how many degrees of freedom (or variables) the model lost due to this linear dependence. Each singular relationship removes the ability to estimate one coefficient uniquely. Therefore, the immediate goal of the analyst is to identify and eliminate the source of this dependency to restore the matrix to full rank.

The Mathematical Core: Why Singularities Halt Estimation

To appreciate why a singular matrix stops the estimation process, one must consider the underlying mathematics of linear modeling. Whether you are using Ordinary Least Squares (OLS) or the iterative Maximum Likelihood Estimation (MLE) used in **GLMs**, the core process involves solving for the vector of coefficients ($beta$) by manipulating the **design matrix** X.

In standard OLS, the solution is given by the formula $beta = (X'X)^{-1}X'y$. The key component here is the matrix product $X'X$ (the matrix of sums of squares and cross-products). The ability to solve for $beta$ hinges entirely on the invertibility of $X'X$. A matrix is singular (non-invertible) if its determinant is exactly zero. When perfect **multicollinearity** exists, the determinant of $X'X$ is zero, and the inverse $(X'X)^{-1}$ cannot be computed.

Although **GLMs** use iterative methods like Fisher Scoring rather than direct OLS inversion, these iterative processes still rely on calculating the inverse of the Hessian matrix (which approximates $X'X$ in the final steps). If the underlying **design matrix** is rank-deficient, the Hessian matrix will also be singular, causing the estimation routine to fail for the correlated term. R then defaults to reporting the coefficient as "not defined" and assigning it an **NA** value.

Reproducing the Error: A Practical R Demonstration

To clearly demonstrate how this error manifests, we will construct a dataset specifically designed to contain perfect linear dependence. We define three predictor variables (x_1, x_2, x_3) where x_2 is simply $2 times x_1$. We then attempt to fit a logistic regression model--a common type of **GLM**--using the **glm() function**.

```
#define data
df <- data.frame(y = c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1),
x1 = c(3, 3, 4, 4, 3, 2, 5, 8, 9, 9, 9, 8, 9, 9, 9),
x2 = c(6, 6, 8, 8, 6, 4, 10, 16, 18, 18, 18, 16, 18, 18, 18),
x3 = c(4, 7, 7, 3, 8, 9, 9, 8, 7, 8, 9, 4, 9, 10, 13))

#fit logistic regression model
model <- glm(y~x1+x2+x3, data=df, family=binomial)

#view model summary
```

summary(model)

Call:
glm(formula = y ~ x1 + x2 + x3, family = binomial, data = df)

Deviance Residuals:
Min 1Q Median 3Q Max
-1.372e-05 -2.110e-08 2.110e-08 2.110e-08 1.575e-05

Coefficients: (1 not defined because of singularities)
Estimate Std. Error z value Pr(>|z|)
(Intercept) -75.496 176487.031 0.000 1
x1 14.546 24314.459 0.001 1
x2 NA NA NA NA
x3 -2.258 20119.863 0.000 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2.0728e+01 on 14 degrees of freedom
Residual deviance: 5.1523e-10 on 12 degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 24

The resulting output clearly displays the expected warning immediately above the coefficient table. More importantly, when reviewing the coefficient estimates, we see definitive proof of the estimation failure: the row corresponding to predictor **x2** contains **NA** values across the board--for the Estimate, Standard Error, z value, and p-value. This confirms that the variable **x2** could not be estimated because it provided no unique information to the model beyond what was already contained in **x1**.

Locating the Problem: Using the Correlation Matrix

While the creation of a redundant variable was intentional in the demonstration above, in real-world applications involving large datasets, the source of the **singularities** may not be obvious. The most reliable and efficient method to diagnose perfect linear dependence is by computing the correlation matrix of the predictor variables using the **cor() function**.

The correlation matrix provides the pairwise correlation coefficient between every variable in the specified data frame. We are specifically looking for coefficients that are exactly **1.0** (perfect positive correlation) or exactly **-1.0** (perfect negative correlation). These exact values confirm the

existence of a deterministic linear relationship that causes the matrix to be singular.

Applying the **cor() function** to our example data frame yields the following results:

#create correlation matrix
cor(df)

y x1 x2 x3
y 1.0000000 0.9675325 0.9675325 0.3610320
x1 0.9675325 1.0000000 1.0000000 0.3872889
x2 0.9675325 1.0000000 1.0000000 0.3872889
x3 0.3610320 0.3872889 0.3872889 1.0000000

The matrix clearly reveals the culprit: the cell corresponding to the intersection of **x1** and **x2** shows a correlation coefficient of precisely **1.0000000**. This diagnostic step confirms that these two variables are perfectly collinear. It is essential to stress the distinction between perfect correlation (1.0) and high correlation (e.g., 0.999). Only the former leads to the singularity error and non-estimable coefficients; high correlation is a separate issue related to estimation instability, not mathematical impossibility.

Implementing the Definitive Solution and Interpretation

Once the perfectly correlated variables are identified, the required correction is simple yet mandatory: one of the redundant predictors must be removed from the **GLM** formula. Since **x1** and **x2** contain identical information in terms of linear prediction, retaining both serves no purpose and prevents successful coefficient estimation.

We modify the **glm() function** call to exclude **x2**, thereby eliminating the linear dependency and restoring the **design matrix** to full rank. This allows the iterative estimation procedure to converge successfully and provide unique coefficient estimates for the remaining variables.

#fit logistic regression model excluding the redundant variable x2
model <- glm(y~x1+x3, data=df, family=binomial)

#view model summary
summary(model)

Call:
glm(formula = y ~ x1 + x3, family = binomial, data = df)

Deviance Residuals:
Min 1Q Median 3Q Max

-1.372e-05 -2.110e-08 2.110e-08 2.110e-08 1.575e-05

Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -75.496 176487.031 0.000 1
x1 14.546 24314.459 0.001 1
x3 -2.258 20119.863 0.000 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2.0728e+01 on 14 degrees of freedom
Residual deviance: 5.1523e-10 on 12 degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 24

As demonstrated by the revised output, the singularity warning is absent. All remaining coefficients (for the intercept, **x1**, and **x3**) now have valid estimates, standard errors, and p-values. The model is mathematically sound and ready for interpretation.

A crucial point for interpretation is that the choice of which variable to drop (**x1** or **x2**) does not affect the model's overall fit statistics, such as the residual deviance, degrees of freedom, or AIC. Since the variables are mathematically interchangeable, they contribute the same predictive power. If we had kept **x2** and dropped **x1**, the coefficient for **x2** would be estimated (at half the magnitude of **x1**'s coefficient, reflecting their 2:1 linear relationship), but the overall predictions would remain identical.

Therefore, the decision should be based on non-statistical criteria, prioritizing **interpretability**--retaining the variable that is more intuitive or standardized within the research field--or **data quality**, favoring the variable with fewer missing values or known measurement precision.

Beyond Simple Duplication: Common Causes of Perfect Collinearity

While the illustrative example used a simple multiplicative relationship, perfect **multicollinearity** frequently arises in more subtle ways, particularly when analysts use derived variables, transformations, or handle categorical factors improperly. Understanding these common pitfalls is essential for proactive model building.

The Dummy Variable Trap: This is arguably the most common cause of **singularities** when using categorical data. If a categorical predictor has K levels, only $K-1$ indicator (dummy) variables should be included in the model alongside the overall intercept. Including all K dummy

variables creates a perfect linear dependency, as the Kth variable is perfectly represented by the absence of the other $K-1$ variables plus the intercept term. While R's native handling often manages this automatically, specifying the full set explicitly will trigger the singularity error.

Inclusion of Component and Sum: If you include several components of a whole along with their total sum, you introduce perfect dependence. For instance, including the counts of "Successful Trials," "Failed Trials," and "Total Trials" in the same model will create a singularity because Total = Success + Failure.

Linear Transformations: Including variables measured in different units that relate linearly, such as age in years and age in months, or temperature in Celsius and temperature in Fahrenheit, will cause perfect **multicollinearity** because one variable is merely a deterministic transformation of the other.

In all these scenarios, the underlying issue remains the same: the **design matrix** is rank-deficient because it contains redundant information. The "not defined because of singularities" warning serves as a mathematically rigorous alert that the model structure must be pruned before reliable estimation can occur. Using the **cor() function** remains the definitive tool for pinpointing the exact relationship, leading directly to an efficient and robust model correction.

Additional Resources for R Error Handling

Mastering the complexities of statistical modeling requires familiarity with handling common technical and mathematical limitations. The singularity error is one challenge; others relate to model convergence and estimation stability. The following topics cover related issues frequently encountered during complex data analysis in R:

Handling non-convergence warnings that occur during the iterative fitting procedures of models like the **GLM**.
Strategies for diagnosing and resolving rank deficiency warnings that arise specifically when manipulating or creating factor variables.
Advanced methods, such as Variance Inflation Factors (VIFs), for detecting and managing high (but not perfect) **multicollinearity**, which causes unstable estimates rather than fatal singularities.