

# Why did I receive the warning “glm.fit: algorithm did not converge” when using the Handle R package?

Authored by  
**stats writer**

May 12, 2024

## RECOMMENDED CITATION

stats writer (2024). *Why did I receive the warning “glm.fit: algorithm did not converge” when using the Handle R package?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=143978>

The warning "glm.fit: algorithm did not converge" is typically received when using the Handle R package due to a failure of the algorithm to reach a stable solution. This warning indicates that the model fitting process was unable to converge and may be the result of various factors such as incorrect data input, inappropriate model specification, or numerical instability. It is important to carefully examine the data and model parameters to ensure accurate and reliable results when using the Handle R package.

## Handle R Warning: glm.fit: algorithm did not converge

One common warning you may encounter in R is:

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

This warning often occurs when you attempt to fit a and you experience perfect separation - that is, a predictor variable is able to perfectly separate the response variable into 0's and 1's.

The following example shows how to handle this warning in practice.

How to Reproduce the Warning

Suppose we attempt to fit the following logistic regression model in R:

```
#create data frame
```

```
df <- data.frame(x=c(.1, .2, .3, .4, .5, .6, .7, .8, .9, 1, 1, 1.1,
```

**1.3, 1.5, 1.7),**

**y=c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1))**

**#attempt to fit logistic regression model**

**glm(y~x, data=df, family="binomial")**

**Call: glm(formula = y ~ x, family = "binomial", data = df)**

**Coefficients:**

**(Intercept) x**

**-409.1 431.1**

**Degrees of Freedom: 14 Total (i.e. Null); 13 Residual**

**Null Deviance: 20.19**

**Residual Deviance: 2.468e-09 AIC: 4**

**Warning messages:**

**1: glm.fit: algorithm did not converge**

**2: glm.fit: fitted probabilities numerically 0 or 1 occurred**

**Notice that we receive the warning message: glm.fit: algorithm did not converge.**

**We receive this message because the predictor variable x is able to perfectly separate the response variable y**

into 0's and 1's.

Notice that for every x value less than 1, y is equal to 0. And for every x value equal to or greater than 1, y is equal to 1.

The following code shows a scenario where the predictor variable is not able to perfectly separate the response variable into 0's and 1's:

```
#create data frame
```

```
df <- data.frame(x=c(.1, .2, .3, .4, .5, .6, .7, .8, .9, 1, 1, 1.1,  
1.3, 1.5, 1.7),  
y=c(0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1))
```

```
#fit logistic regression model
```

```
glm(y~x, data=df, family="binomial")
```

```
Call: glm(formula = y ~ x, family = "binomial", data = df)
```

```
Coefficients:
```

```
(Intercept) x
```

```
-2.112 2.886
```

```
Degrees of Freedom: 14 Total (i.e. Null); 13 Residual
```

```
Null Deviance: 20.73
```

**Residual Deviance: 16.31 AIC: 20.31**

**We don't receive any warning message because the predictor variable is not able to perfectly separate the response variable into 0's and 1's.**

**How to Handle the Warning**

**If we encounter a scenario with perfect separation, there are two ways to handle it:**

**Method 1: Use penalized regression.**

**One option is to use some form of penalized logistic regression such as lasso logistic regression or elastic-net regularization.**

**Method 2: Use the predictor variable to perfectly predict the response variable.**

**If you suspect that this perfect separation may exist in the population, you can simply use that predictor variable to perfectly predict the value of the response variable.**

**For example, in the above scenario we saw that the**

**response variable  $y$  was always equal to 0 when the predictor variable  $x$  was less than 1.**

**If we suspect that this relationship holds in the overall population, we can just always predict that the value of  $y$  will be equal to 0 when  $x$  is less than 1 and not worry about fitting some penalized logistic regression model.**

**The following tutorials offer additional information on working with the `glm()` function in R:**

ARABPSYCHOLOGY.COM