How can weighted least squares regression be performed in R?

Authored by

stats writer

April 24, 2024

Weighted least squares regression is a statistical method used in data analysis to fit a linear model to a dataset. It is particularly useful when dealing with data that has unequal variances or heteroscedasticity. In R, this method can be performed by using the "lm" function and specifying the weights argument as a vector of weights corresponding to each data point. The weights are used to give more importance to certain data points, resulting in a more accurate and robust regression model. Additionally, the "lm" function also allows for the specification of different types of weights, such as inverse variance weights or user-defined weights. Overall, performing weighted least squares regression in R allows for the analysis of data with varying levels of uncertainty and can lead to more reliable and accurate results.

Perform Weighted Least Squares Regression in R

One of the key is that the are distributed with equal variance at each level of the predictor variable. This assumption is known as homoscedasticity.

When this assumption is violated, we say that is present in the residuals. When this occurs, the results of the regression become unreliable.

One way to handle this issue is to instead use weighted least squares regression, which places weights on the such that those with small error variance are given more weight since they contain more information compared to observations with larger error variance.

This tutorial provides a step-by-step example of how to perform weight least squares regression in R.

Step 1: Create the Data

The following code creates a data frame that contains the number of hours studied and the corresponding exam score for 16 students:

```
df <- data.frame(hours=c(1, 1, 2, 2, 2, 3, 4, 4, 4, 5, 5, 5, 6, 6, 7, 8),
score=c(48, 78, 72, 70, 66, 92, 93, 75, 75, 80, 95, 97, 90, 96, 99, 99))
```

Step 2: Perform Linear Regression

Next, we'll use the lm() function to fit a that uses hours as the predictor variable and score as the :

```
#fit simple linear regression model
model <- lm(score ~ hours, data = df)

#view summary of model
summary(model)

Call:
lm(formula = score ~ hours, data = df)

Residuals:
Min 1Q Median 3Q Max
```

-17.967 -5.970 -0.719 7.531 15.032

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 60.467 5.128 11.791 1.17e-08 ***
hours 5.500 1.127 4.879 0.000244 ***

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

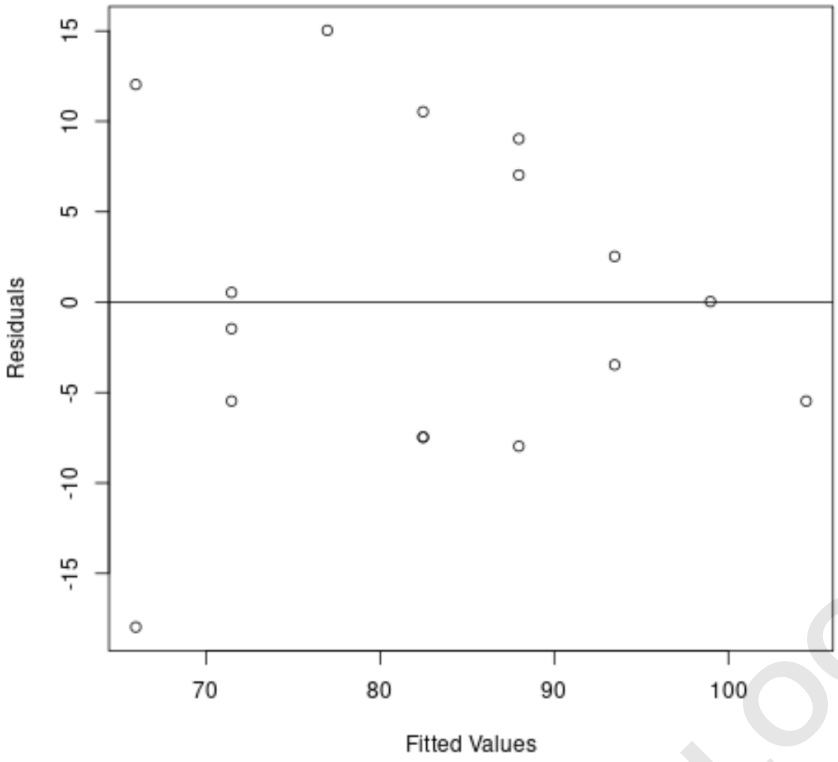
Residual standard error: 9.224 on 14 degrees of freedom
Multiple R-squared: 0.6296, Adjusted R-squared: 0.6032
F-statistic: 23.8 on 1 and 14 DF, p-value: 0.0002438

Step 3: Test for Heteroscedasticity

Next, we'll create a residual vs. fitted values plot to visually check for heteroscedasticity:

```
#create residual vs. fitted plot
plot(fitted(model), resid(model), xlab='Fitted Values', ylab='Residuals')

#add a horizontal line at 0
abline(0,0)
```



We can see from the plot that the residuals exhibit a "cone" shape - they're not distributed with equal variance throughout the plot.

To formally test for heteroscedasticity, we can perform a Breusch-Pagan test:

```
#load lmtest package
library(lmtest)

#perform Breusch-Pagan test
bptest(model)
```

studentized Breusch-Pagan test

data: model
BP = 3.9597, df = 1, p-value = 0.0466

Null Hypothesis (H0): Homoscedasticity is present (the residuals are distributed with equal variance)Alternative Hypothesis (HA): Heteroscedasticity is present (the residuals are not distributed with equal variance)

Since the p-value from the test is 0.0466 we will reject the null hypothesis and conclude that heteroscedasticity is a problem in this model.

Step 4: Perform Weighted Least Squares Regression

Since heteroscedasticity is present, we will perform weighted least squares by defining the weights in such a way that the observations with lower variance are given more weight:

```
#define weights to use
wt <- 1 / lm(abs(model$residuals) ~ model$fitted.values)$fitted.values^2

#perform weighted least squares regression
```

```
wls_model <- lm(score ~ hours, data = df, weights=wt)

#view summary of model
summary(wls_model)

Call:
lm(formula = score ~ hours, data = df, weights = wt)

Weighted Residuals:
Min 1Q Median 3Q Max
-2.0167 -0.9263 -0.2589 0.9873 1.6977

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 63.9689 5.1587 12.400 6.13e-09 ***
hours 4.7091 0.8709 5.407 9.24e-05 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.199 on 14 degrees of freedom
Multiple R-squared: 0.6762, Adjusted R-squared: 0.6531
F-statistic: 29.24 on 1 and 14 DF, p-value: 9.236e-05
```

From the output we can see that the coefficient estimate

for the predictor variable *hours* changed a bit and the overall fit of the model improved.

The weighted least squares model has a residual standard error of 1.199 compared to 9.224 in the original simple linear regression model.

This indicates that the predicted values produced by the weighted least squares model are much closer to the actual observations compared to the predicted values produced by the simple linear regression model.

The weighted least squares model also has an R-squared of .6762 compared to .6296 in the original simple linear regression model.

This indicates that the weighted least squares model is able to explain more of the variance in exam scores compared to the simple linear regression model.

These metrics indicate that the weighted least squares model offers a better fit to the data compared to the simple linear regression model.