

How can data be transformed in R using methods such as logarithm, square root, and cube root?

Authored by
stats writer

April 20, 2024

RECOMMENDED CITATION

stats writer (2024). *How can data be transformed in R using methods such as logarithm, square root, and cube root?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137473>

Data transformation in R refers to the process of manipulating or changing the values of a dataset in order to make it more suitable for analysis or to meet specific requirements. This can be done using various methods, such as logarithm, square root, and cube root.

The logarithm function in R allows us to transform data that is highly skewed or has a wide range of values into a more evenly distributed form. This is particularly useful when dealing with data that follows an exponential or power law distribution. Taking the square root or cube root of a dataset can also help in reducing the effect of extreme values, making the data more normally distributed. These transformations can also help in improving the linear relationship between variables and making it easier to apply certain statistical tests. Overall, data transformation in R using these methods can help in improving the accuracy and interpretability of the results obtained from data analysis.

Transform Data in R (Log, Square Root, Cube Root)

Many statistical tests make the assumption that the residuals of a response variable are normally distributed.

However, often the residuals are *not* normally distributed. One way to address this issue is to transform the response variable using one of the three transformations:

1. Log Transformation: Transform the response variable from y to $\log(y)$.

2. Square Root Transformation: Transform the response variable from y to \sqrt{y} .

3. Cube Root Transformation: Transform the response

variable from y to $y^{1/3}$.

By performing these transformations, the response variable typically becomes closer to normally distributed. The following examples show how to perform these transformations in R.

Log Transformation in R

The following code shows how to perform a log transformation on a response variable:

```
#create data frame
```

```
df <- data.frame(y=c(1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 6, 7, 8),  
x1=c(7, 7, 8, 3, 2, 4, 4, 6, 6, 7, 5, 3, 3, 5, 8),  
x2=c(3, 3, 6, 6, 8, 9, 9, 8, 8, 7, 4, 3, 3, 2, 7))
```

```
#perform log transformation
```

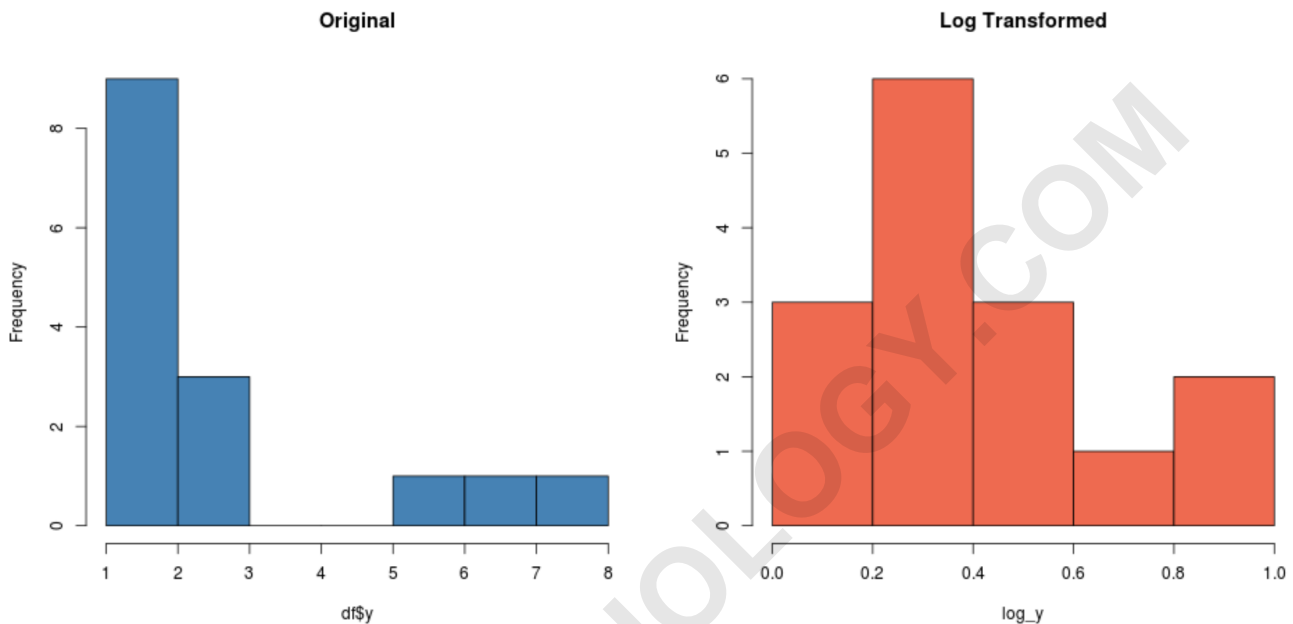
```
log_y <- log10(df$y)
```

The following code shows how to create histograms to view the distribution of y before and after performing a log transformation:

```
#create histogram for original distribution
```

```
hist(df$y, col='steelblue', main='Original')
```

```
#create histogram for log-transformed distribution  
hist(log_y, col='coral2', main='Log Transformed')
```



Notice how the log-transformed distribution is much more normal compared to the original distribution. It's still not a perfect "bell shape" but it's closer to a normal distribution than the original distribution.

In fact, if we perform a Shapiro-Wilk test on each distribution we'll find that the original distribution fails the normality assumption while the log-transformed distribution does not (at $\alpha = .05$):

```
#perform Shapiro-Wilk Test on original data
```

```
shapiro.test(df$y)
```

Shapiro-Wilk normality test

```
data: df$y
```

```
W = 0.77225, p-value = 0.001655
```

```
#perform Shapiro-Wilk Test on log-transformed data
```

```
shapiro.test(log_y)
```

Shapiro-Wilk normality test

```
data: log_y
```

```
W = 0.89089, p-value = 0.06917
```

Square Root Transformation in R

The following code shows how to perform a square root transformation on a response variable:

```
#create data frame
```

```
df <- data.frame(y=c(1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 6, 7, 8),
```

```
x1=c(7, 7, 8, 3, 2, 4, 4, 6, 6, 7, 5, 3, 3, 5, 8),
```

```
x2=c(3, 3, 6, 6, 8, 9, 9, 8, 8, 7, 4, 3, 3, 2, 7))
```

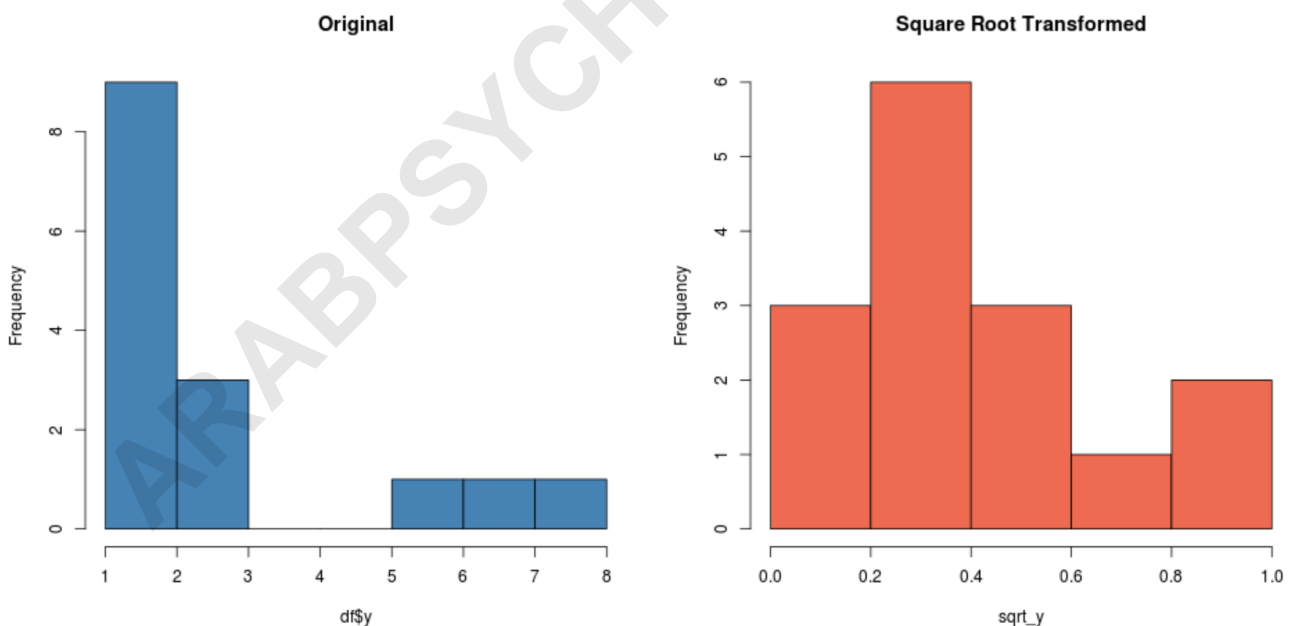
```
#perform square root transformation
```

```
sqrt_y <- sqrt(df$y)
```

The following code shows how to create histograms to view the distribution of y before and after performing a square root transformation:

```
#create histogram for original distribution  
hist(df$y, col='steelblue', main='Original')
```

```
#create histogram for square root-transformed  
distribution  
hist(sqrt_y, col='coral2', main='Square Root  
Transformed')
```



Notice how the square root-transformed distribution is much more normally distributed compared to the

original distribution.

Cube Root Transformation in R

The following code shows how to perform a cube root transformation on a response variable:

```
#create data frame
```

```
df <- data.frame(y=c(1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 6, 7, 8),  
x1=c(7, 7, 8, 3, 2, 4, 4, 6, 6, 7, 5, 3, 3, 5, 8),  
x2=c(3, 3, 6, 6, 8, 9, 9, 8, 8, 7, 4, 3, 3, 2, 7))
```

```
#perform square root transformation
```

```
cube_y <- df$y^(1/3)
```

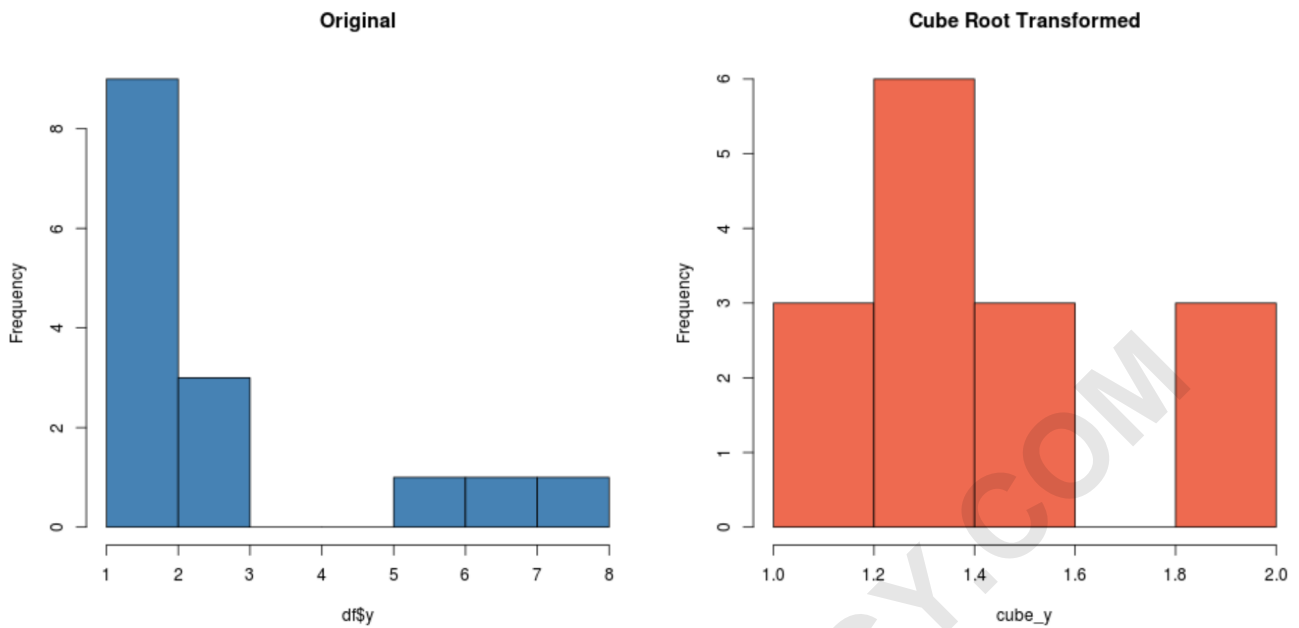
The following code shows how to create histograms to view the distribution of y before and after performing a square root transformation:

```
#create histogram for original distribution
```

```
hist(df$y, col='steelblue', main='Original')
```

```
#create histogram for square root-transformed  
distribution
```

```
hist(cube_y, col='coral2', main='Cube Root  
Transformed')
```



Depending on your dataset, one of these transformations may produce a new dataset that is more normally distributed than the others.