

How can I create a plot of predicted values in R, and what are some examples of how to do so?

Authored by
stats writer

April 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a plot of predicted values in R, and what are some examples of how to do so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=140429>

Creating a plot of predicted values in R is a useful way to visualize and analyze data. This can be done using various functions and packages in R, such as the "predict" function and the "ggplot2" package. To create a plot of predicted values, you first need to fit a model to your data. This can be done using regression, classification, or other statistical methods. Once the model is fitted, the "predict" function can be used to generate predicted values based on the model. These predicted values can then be plotted using the "ggplot2" package, which offers a variety of customizable plot types and features. Some examples of creating a plot of predicted values in R include visualizing the relationship between a predictor variable and the response variable, comparing predicted values to actual values, and examining the performance of different models. Overall, creating a plot of predicted values in R can provide valuable insights and aid in the interpretation of data.

Plot Predicted Values in R (With Examples)

Often you may want to plot the predicted values of a regression model in R in order to visualize the differences between the predicted values and the actual values.

This tutorial provides examples of how to create this type of plot in base R and ggplot2.

Example 1: Plot of Predicted vs. Actual Values in Base R

The following code shows how to fit a in R and then create a plot of predicted vs. actual values:

```
#create data
```

```
df <- data.frame(x1=c(3, 4, 4, 5, 5, 6, 7, 8, 11, 12),  
x2=c(6, 6, 7, 7, 8, 9, 11, 13, 14, 14),
```

```
y=c(22, 24, 24, 25, 25, 27, 29, 31, 32, 36))
```

```
#fit multiple linear regression model
```

```
model <- lm(y ~ x1 + x2, data=df)
```

```
#plot predicted vs. actual values
```

```
plot(x=predict(model), y=df$y,
```

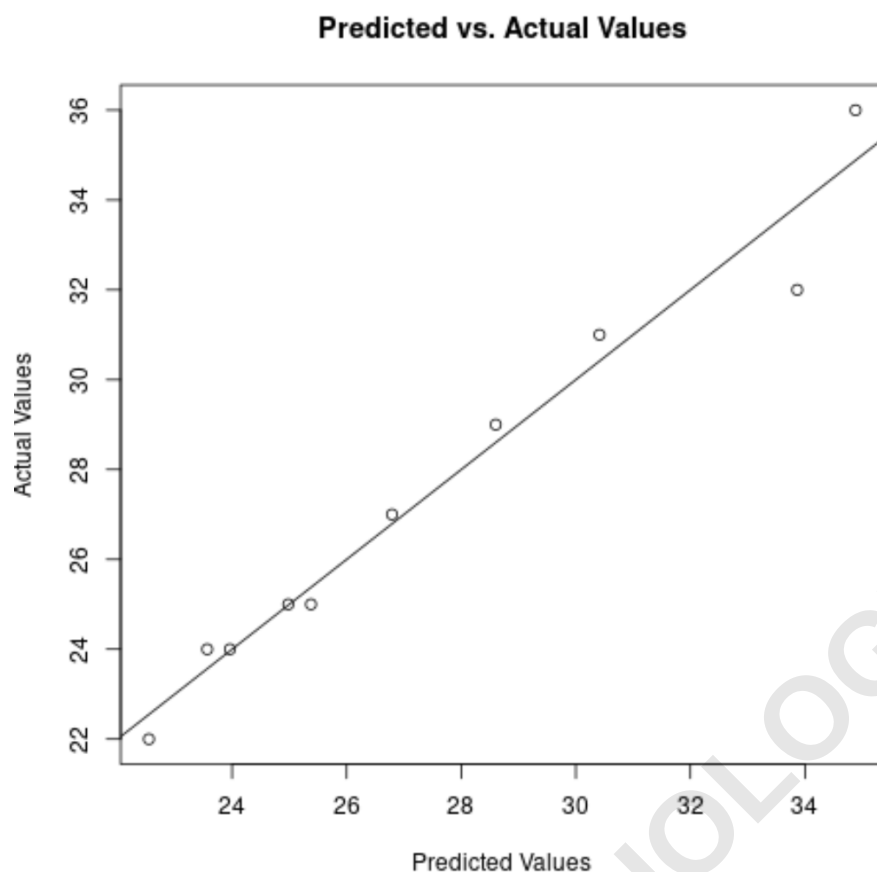
```
xlab='Predicted Values',
```

```
ylab='Actual Values',
```

```
main='Predicted vs. Actual Values')
```

```
#add diagonal line for estimated regression line
```

```
abline(a=0, b=1)
```



The x-axis displays the predicted values from the model and the y-axis displays the actual values from the dataset. The diagonal line in the middle of the plot is the estimated regression line.

Since each of the data points lies fairly close to the estimated regression line, this tells us that the regression model does a pretty good job of fitting the data.

We can also create a data frame that shows the actual

and predicted values for each data point:

```
#create data frame of actual and predicted values  
values <- data.frame(actual=df$y,  
predicted=predict(model))
```

```
#view data frame  
values
```

```
actual predicted
```

```
1 22 22.54878
```

```
2 24 23.56707
```

```
3 24 23.96341
```

```
4 25 24.98171
```

```
5 25 25.37805
```

```
6 27 26.79268
```

```
7 29 28.60366
```

```
8 31 30.41463
```

```
9 32 33.86585
```

```
10 36 34.88415
```

Example 2: Plot of Predicted vs. Actual Values in ggplot2

The following code shows how to create a plot of predicted vs. actual values using the data visualization

package:

library(ggplot2)

#create data

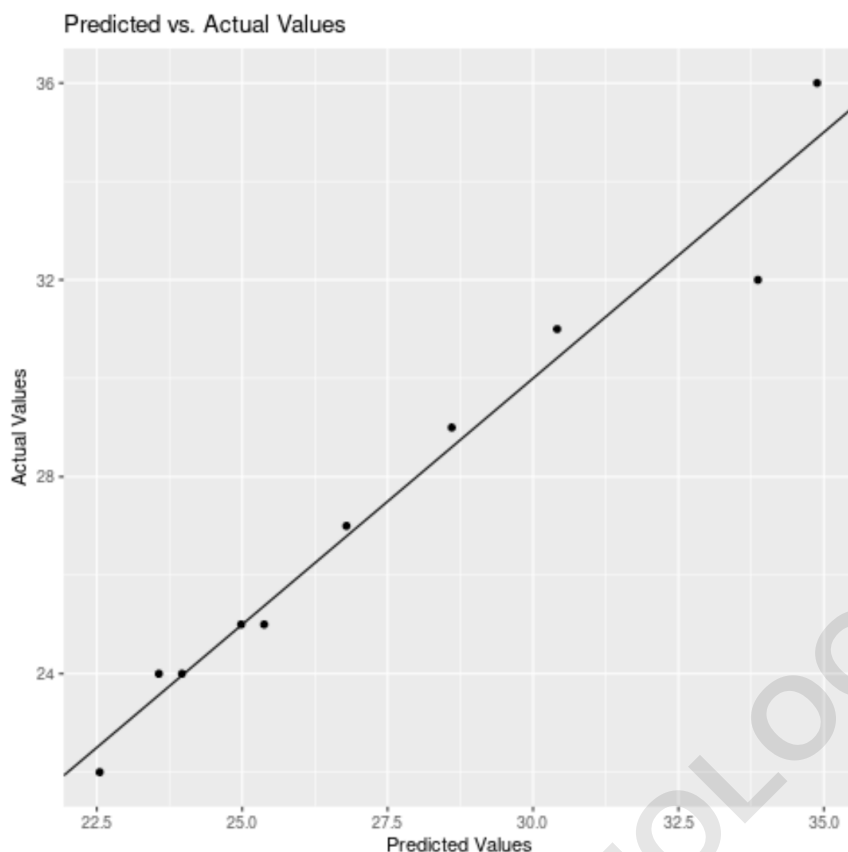
```
df <- data.frame(x1=c(3, 4, 4, 5, 5, 6, 7, 8, 11, 12),  
x2=c(6, 6, 7, 7, 8, 9, 11, 13, 14, 14),  
y=c(22, 24, 24, 25, 25, 27, 29, 31, 32, 36))
```

#fit multiple linear regression model

```
model <- lm(y ~ x1 + x2, data=df)
```

#plot predicted vs. actual values

```
ggplot(df, aes(x=predict(model), y=y)) +  
geom_point() +  
geom_abline(intercept=0, slope=1) +  
labs(x='Predicted Values', y='Actual Values',  
title='Predicted vs. Actual Values')
```



Once again, the x-axis displays the predicted values from the model and the y-axis displays the actual values from the dataset.