

# How can I plot a linear regression line in ggplot2?

Authored by  
**stats writer**

April 20, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I plot a linear regression line in ggplot2?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137494>

Linear regression is a statistical method used to model the relationship between two or more variables. In the context of data visualization, a linear regression line can be plotted to visually represent the trend or pattern in the data. In ggplot2, a popular data visualization package in R, this can be accomplished by first creating a scatterplot of the data and then adding a "geom\_smooth" layer with the "method" argument set to "lm" to indicate linear regression. This will plot a line of best fit through the data points, providing insight into the overall trend and strength of the relationship between the variables. By incorporating this feature in ggplot2, analysts can effectively communicate the results of their linear regression analysis in a clear and concise manner.

## Plot a Linear Regression Line in ggplot2 (With Examples)

You can use the R visualization library **ggplot2** to plot a fitted linear regression model using the following basic syntax:

```
ggplot(data,aes(x, y)) +  
geom_point() +  
geom_smooth(method='lm')
```

The following example shows how to use this syntax in practice.

Example: Plot a Linear Regression Line in ggplot2

Suppose we fit a simple linear regression model to the following dataset:

```
#create dataset
```

```
data <- data.frame(y=c(6, 7, 7, 9, 12, 13, 13, 15, 16, 19,
22, 23, 23, 25, 26),
x=c(1, 2, 2, 3, 4, 4, 5, 6, 6, 8, 9, 9, 11, 12, 12))
```

```
#fit linear regression model to dataset and view model
summary
```

```
model <- lm(y~x, data=data)
summary(model)
```

**Call:**

```
lm(formula = y ~ x, data = data)
```

**Residuals:**

```
Min 1Q Median 3Q Max
-1.4444 -0.8013 -0.2426 0.5978 2.2363
```

**Coefficients:**

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.20041 0.56730 7.404 5.16e-06 ***
x 1.84036 0.07857 23.423 5.13e-12 ***
```

---

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

```
Residual standard error: 1.091 on 13 degrees of
freedom
```

**Multiple R-squared: 0.9769, Adjusted R-squared: 0.9751  
F-statistic: 548.7 on 1 and 13 DF, p-value: 5.13e-12**

The following code shows how to visualize the fitted linear regression model:

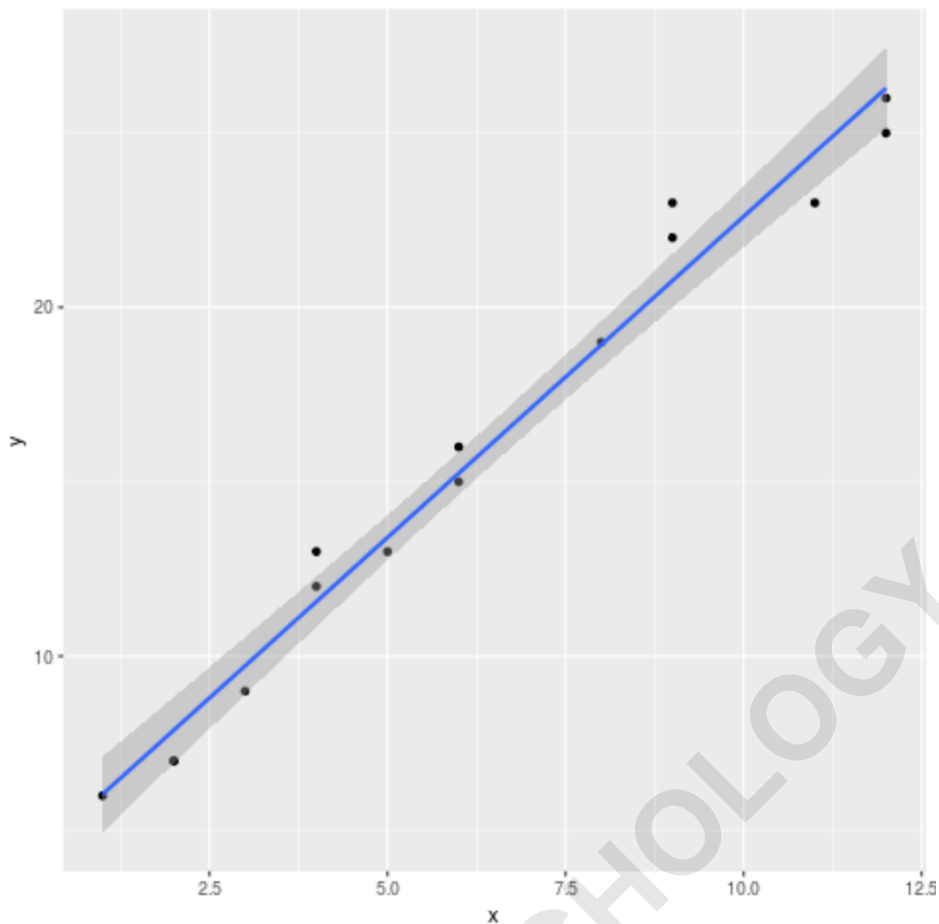
```
library(ggplot2)
```

```
#create plot to visualize fitted linear regression model
```

```
ggplot(data,aes(x, y)) +
```

```
geom_point() +
```

```
geom_smooth(method='lm')
```



**By default, ggplot2 adds standard error lines to the chart. You can disable these by using the argument `se=FALSE` as follows:**

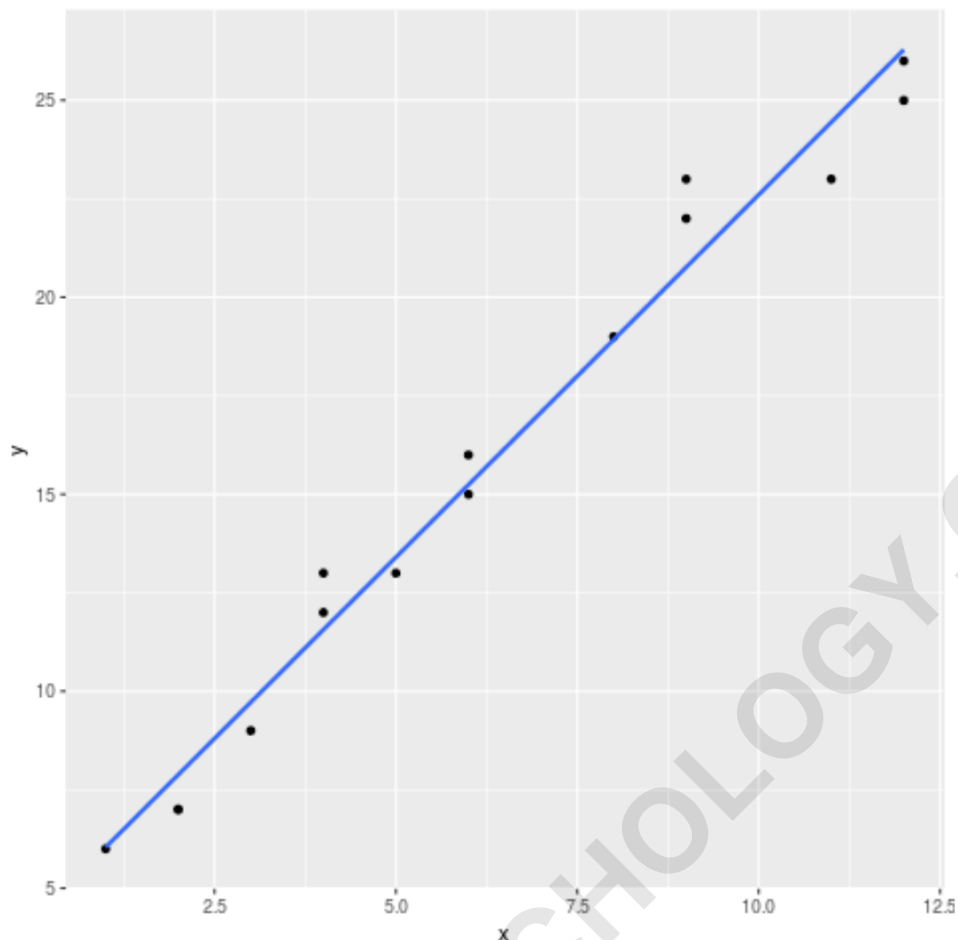
```
library(ggplot2)
```

```
#create regression plot with no standard error lines
```

```
ggplot(data,aes(x, y)) +
```

```
geom_point() +
```

```
geom_smooth(method='lm', se=FALSE)
```



Lastly, we can customize some aspects of the chart to make it more visually appealing:

```
library(ggplot2)
```

```
#create regression plot with customized style
```

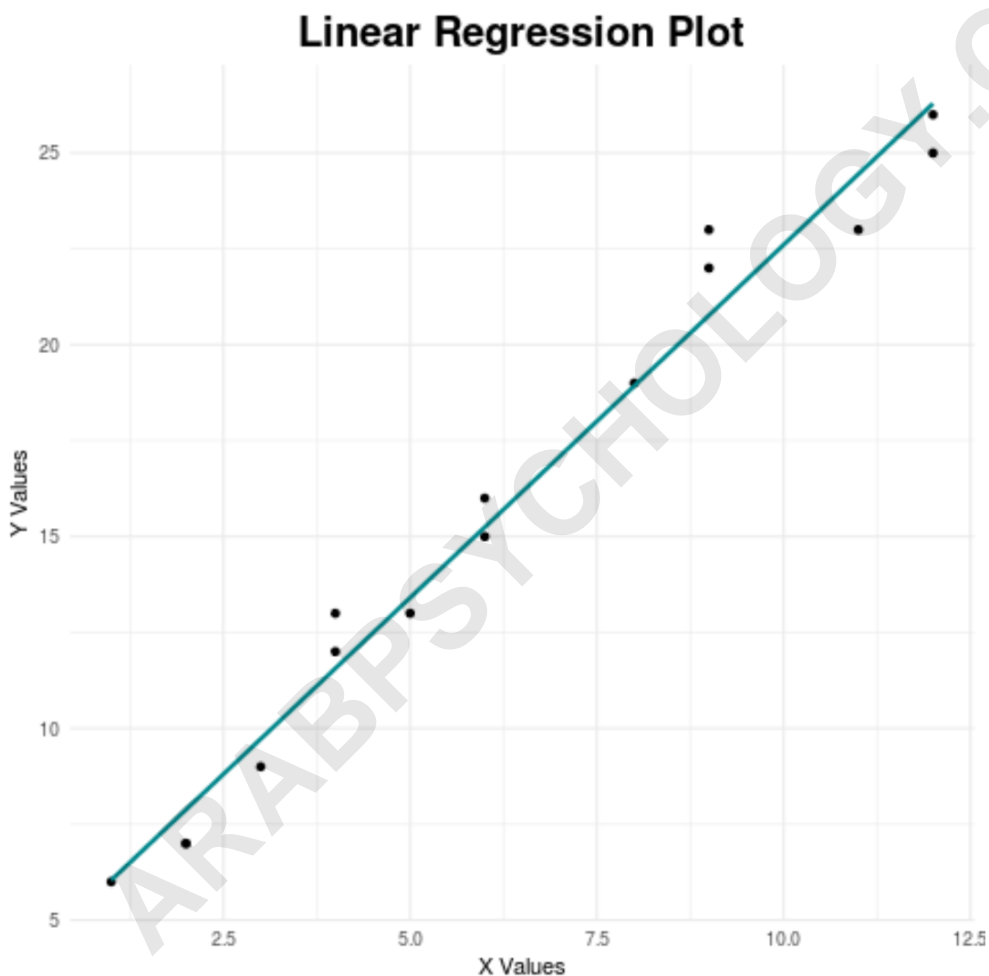
```
ggplot(data,aes(x, y)) +
```

```
geom_point() +
```

```
geom_smooth(method='lm', se=FALSE,
```

```
color='turquoise4') +
```

```
theme_minimal() +  
labs(x='X Values', y='Y Values', title='Linear Regression  
Plot') +  
theme(plot.title = element_text(hjust=0.5, size=20,  
face='bold'))
```



***Refer to [this post](#) for a complete guide to the best ggplot2 themes.***

---

## **An Introduction to Multiple Linear Regression in R** **How to Plot a Confidence Interval in R**

ARABPSYCHOLOGY.COM