How can I plot the results of the lm() function in R?

Authored by

stats writer

June 30, 2024

The lm() function in R is used for fitting linear models to data. It allows users to analyze the relationship between variables and make predictions based on the data. To better understand and interpret the results of the lm() function, it is often helpful to plot the data and the fitted model. This can be done using various graphical tools and packages in R, such as the base plot function or the ggplot2 package. Plotting the results of the lm() function can provide a visual representation of the relationship between variables and aid in the interpretation of the model's performance. This can be particularly useful in identifying patterns, outliers, and potential areas for improvement in the model. By utilizing appropriate plotting techniques, users can effectively communicate and present their findings from the lm() function in a clear and concise manner.

Plot lm() Results in R

You can use the following methods to plot the results of the lm() function in R:

Method 1: Plot lm() Results in Base R

```
#create scatterplot
plot(y ~ x, data=data)

#add fitted regression line to scatterplot
abline(fit)
```

Method 2: Plot lm() Results in ggplot2

```
library(ggplot2)

#create scatterplot with fitted regression line
ggplot(data, aes(x = x, y = y)) +
```

```
geom_point() +
stat_smooth(method = "lm")
```

The following examples shows how to use each method in practice with the built-in in R.

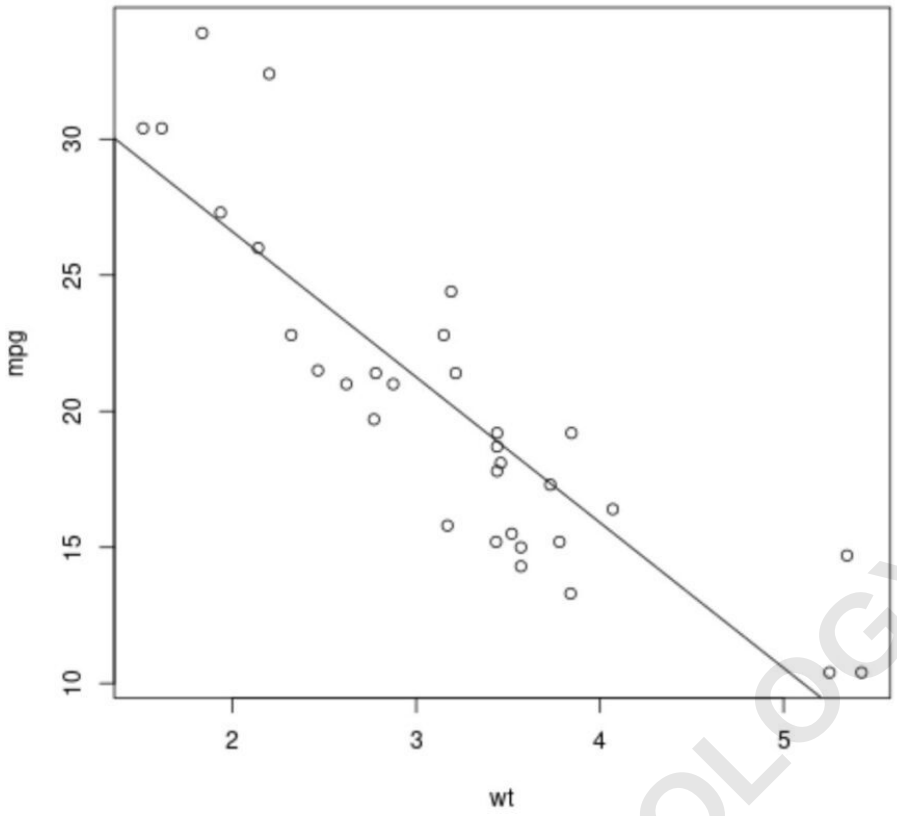
Example 1: Plot lm() Results in Base R

The following code shows how to plot the results of the lm() function in base R:

```
#fit regression model
fit <- lm(mpg ~ wt, data=mtcars)

#create scatterplot
plot(mpg ~ wt, data=mtcars)

#add fitted regression line to scatterplot
abline(fit)
```



The points in the plot represent the raw data values and the straight diagonal line represents the fitted regression line.

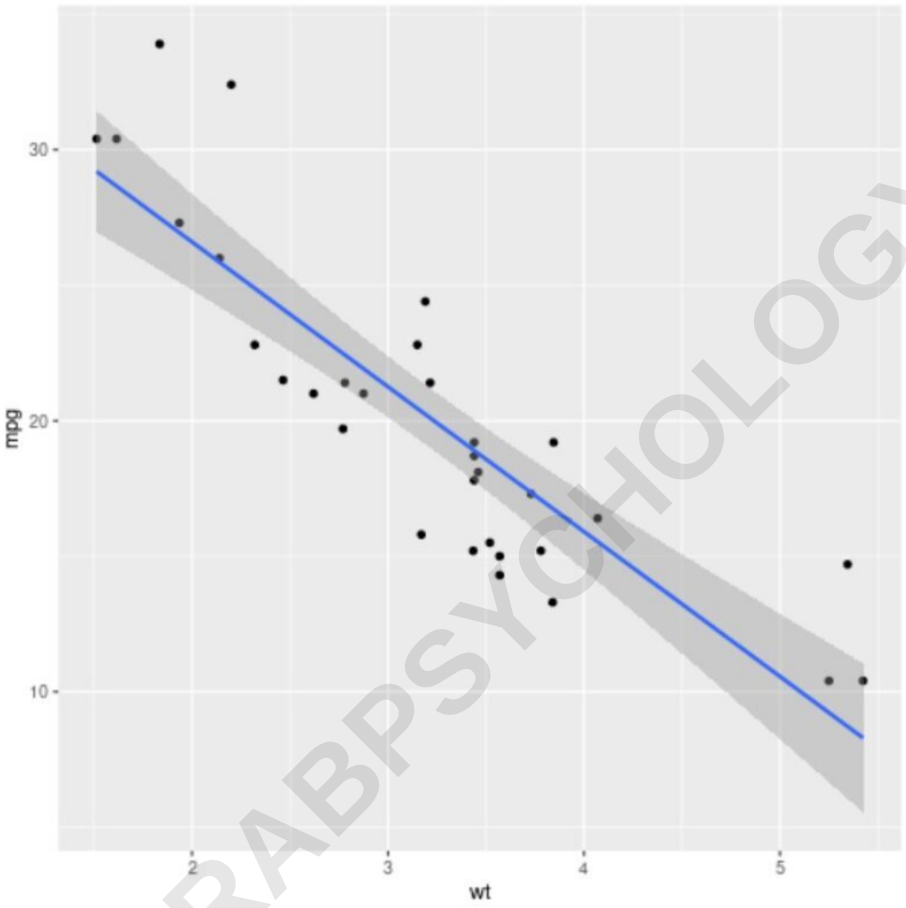
Example 2: Plot lm() Results in ggplot2

The following code shows how to plot the results of the lm() function using the data visualization package:

```
library(ggplot2)

#fit regression model
fit <- lm(mpg ~ wt, data=mtcars)
```

```
#create scatterplot with fitted regression line
ggplot(mtcars, aes(x = x, y = y)) +
geom_point() +
stat_smooth(method = "lm")
```



The blue line represents the fitted regression line and the grey bands represent the 95% confidence interval limits.

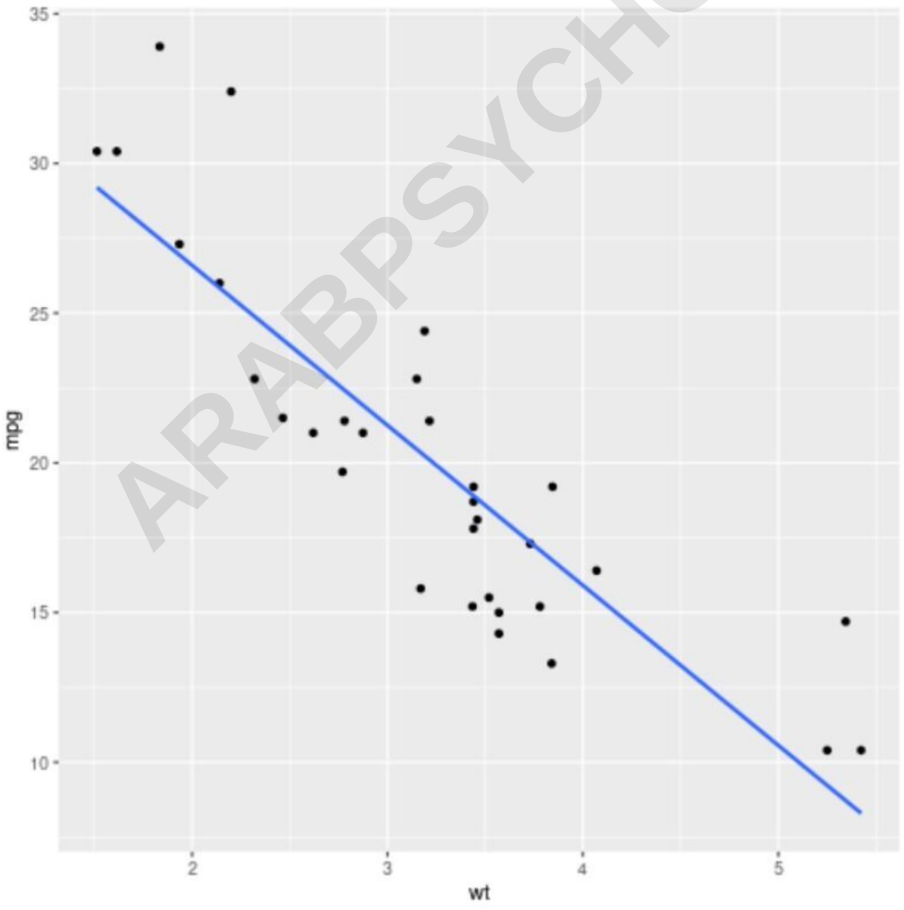
To remove the confidence interval limits, simply use

se=FALSE in the stat_smooth() argument:

```
library(ggplot2)

#fit regression model
fit <- lm(mpg ~ wt, data=mtcars)

#create scatterplot with fitted regression line
ggplot(mtcars, aes(x = x, y = y)) +
geom_point() +
stat_smooth(method = "lm", se=FALSE)
```

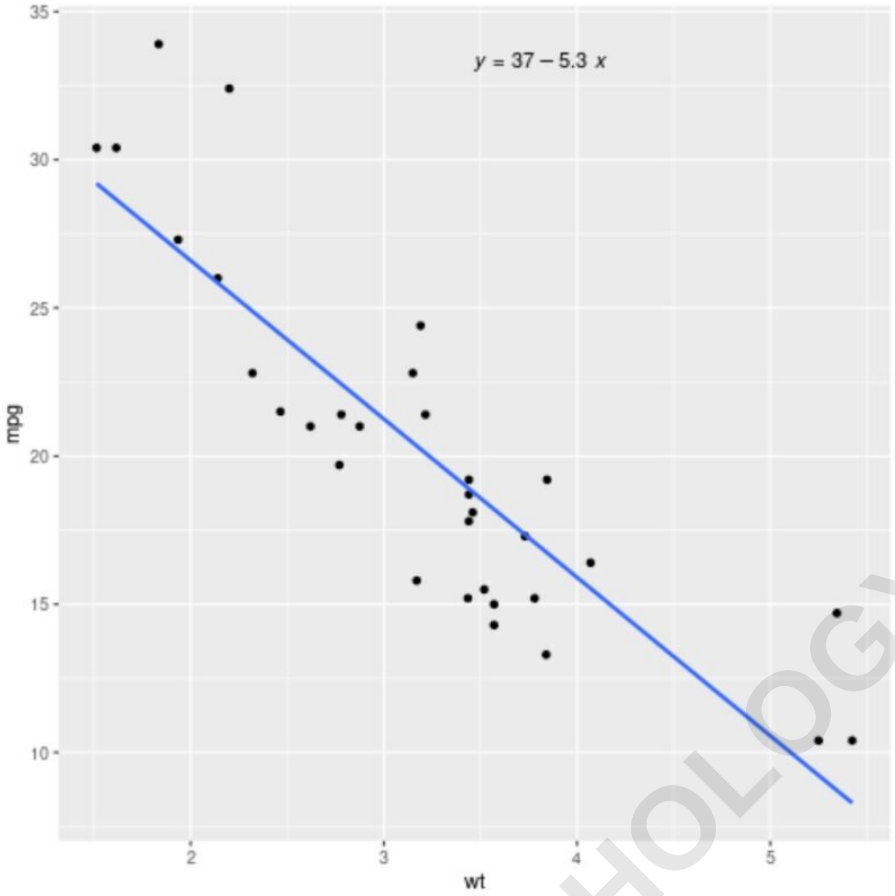


You can also add the fitted regression equation inside the chart by using the stat_regline_equation() function from the ggpubr package:

```
library(ggplot2)
library(ggpubr)

#fit regression model
fit <- lm(mpg ~ wt, data=mtcars)

#create scatterplot with fitted regression line
ggplot(mtcars, aes(x = x, y = y)) +
geom_point() +
stat_smooth(method = "lm", se=FALSE) +
stat_regline_equation(label.x.npc = "center")
```



Additional Resources

The following tutorials explain how to perform other common tasks in R: