Introduction to Linear Regression in R

Ziyang Lyu & Steefan Contractor

2022-10-26

Example

For this analysis, we will use the `cars` dataset that comes with R by default.

`cars` is a standard built-in dataset, that makes it convenient to show linear regression in a simple and easy to understand fashion.

You can access this dataset by typing in `cars` in your R console.

You will find that it consists of 50 observations(rows) and 2 variables (columns) `dist` and `speed.` Lets print out the first six observations here.

```
head(cars)  # display the first 6 observations
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

The goal here is to establish a mathematical equation for `dist` as a function of `speed`, so you can use it to predict `dist` when only the `speed` of the car is known.

So it is desirable to build a linear regression model with the response variable as `dist` and the predictor as `speed`.

Before we begin building the regression model, it is a good practice to analyse and understand the variables.

The graphical analysis and correlation study below will help with this.

Graphical Analysis

The aim of this exercise is to build a simple regression model that you can use to predict Distance (`dist`).

This is possible by establishing a mathematical formula between Distance (`dist`) and Speed (`speed`).

But before jumping in to the syntax, lets try to understand these variables graphically.

Typically, for each of the predictors, the following plots help visualise the patterns:

1. Scatter plot: Visualise the linear relationship between the predictor and response
2. Box plot: To spot any outlier observations in the variable. Having outliers in your predictor can drastically affect the predictions as they can affect the direction/slope of the line of best fit.

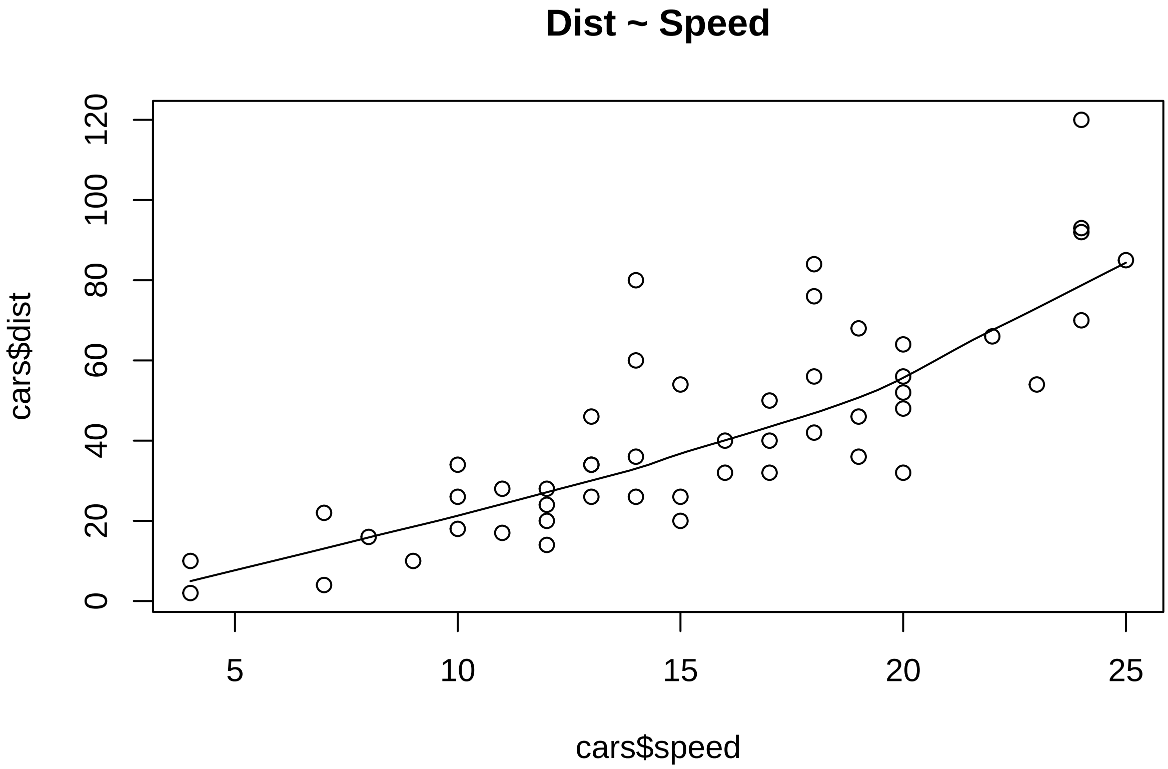
3. Density plot: To see the distribution of the predictor variable. Ideally, a close to normal distribution (a bell shaped curve), without being skewed to the left or right is preferred.

Using Scatter Plot To Visualise The Relationship

Scatter plots can help visualise linear relationships between the response and predictor variables.

Ideally, if you have many predictor variables, a scatter plot is drawn for each one of them against the response, along with the line of best fit as seen below.

```
scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")  # scatterplot
```



The scatter plot along with the smoothing line above suggests a linear and positive relationship between the `dist` and `speed`.

This is a good thing.

Because, one of the underlying assumptions of linear regression is, the relationship between the response and predictor variables is linear and additive.

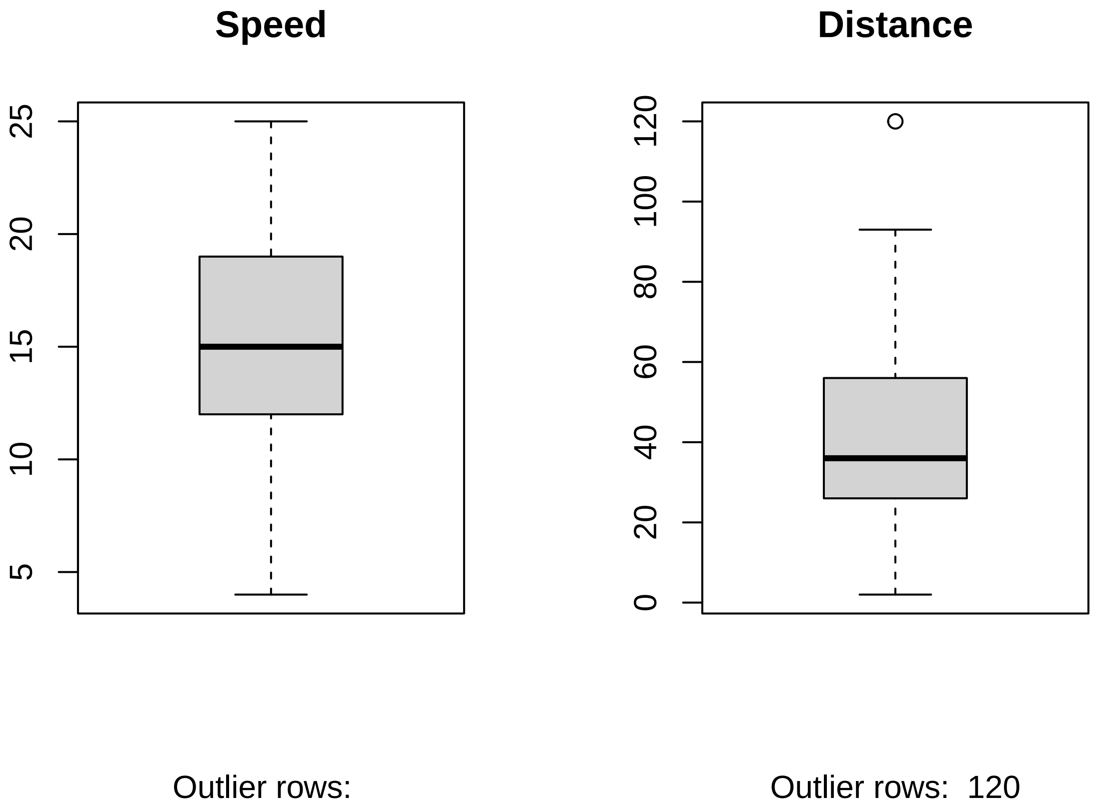
Using BoxPlot To Check For Outliers

Generally, an outlier is any datapoint that lies outside the 1.5 * inter quartile range (IQR).

IQR is calculated as the distance between the 25th percentile and 75th percentile values for that variable.

```
par(mfrow=c(1, 2))  # divide graph area in 2 columns

boxplot(cars$speed, main="Speed", sub=paste("Outlier rows: ", boxplot.stats(cars$speed)$out))  # box pl
boxplot(cars$dist, main="Distance", sub=paste("Outlier rows: ", boxplot.stats(cars$dist)$out))  # box p
```



Using Density Plot To Check If Response Variable Is Close To Normal

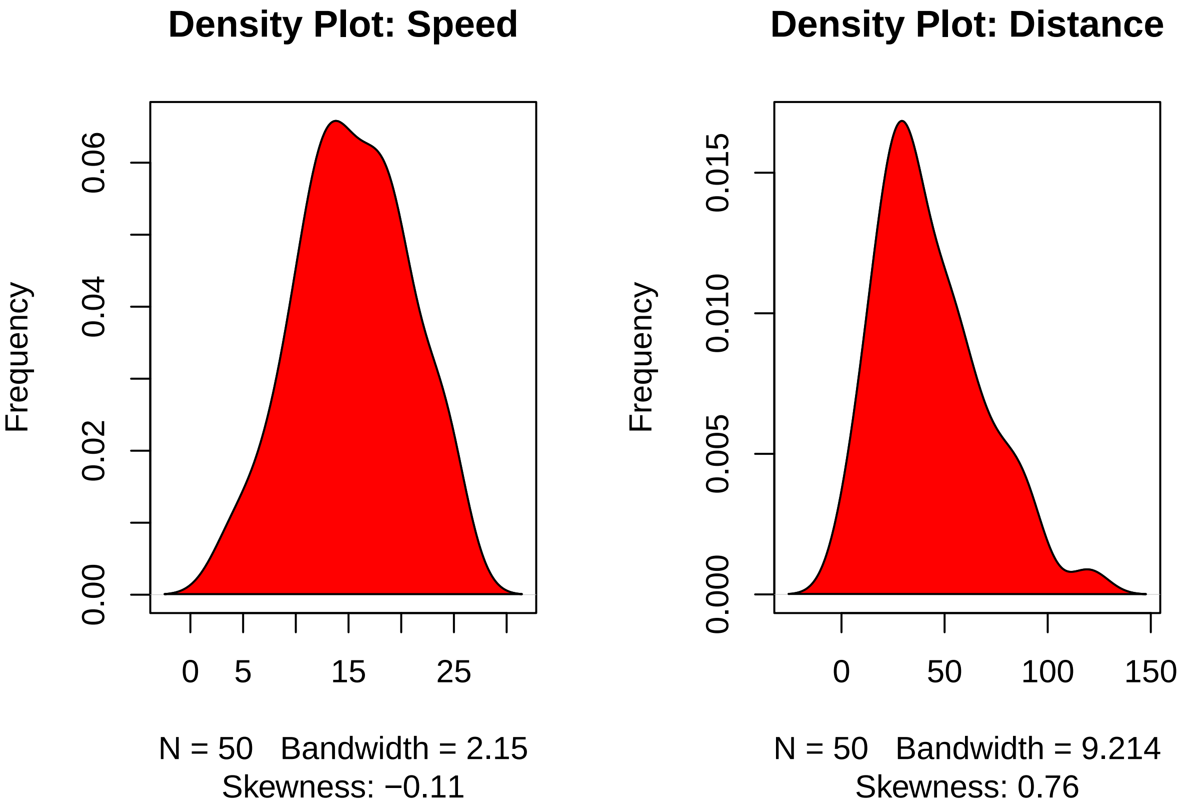
```
library(e1071)  # for skewness function
par(mfrow=c(1, 2))  # divide graph area in 2 columns

plot(density(cars$speed), main="Density Plot: Speed", ylab="Frequency", sub=paste("Skewness:", round(e10

polygon(density(cars$speed), col="red")

plot(density(cars$dist), main="Density Plot: Distance", ylab="Frequency", sub=paste("Skewness:", round(

polygon(density(cars$dist), col="red")
```



What is Correlation Analysis

Correlation analysis studies the strength of relationship between two continuous variables. It involves computing the correlation coefficient between the the two variables.

So what is correlation? And how is it helpful in linear regression?

Correlation is a statistical measure that shows the degree of linear dependence between two variables.

In order to compute correlation, the two variables must occur in pairs, just like what we have here with `speed` and `dist`.

Correlation can take values between -1 to +1.

If one variables consistently increases with increasing value of the other, then they have a strong positive correlation (value close to +1).

Similarly, if one consistently decreases when the other increase, they have a strong negative correlation (value close to -1).

A value closer to 0 suggests a weak relationship between the variables.

A low correlation ($-0.2 < x < 0.2$) probably suggests that much of variation of the response variable (Y) is unexplained by the predictor (X). In that case, you should probably look for better explanatory variables.

If you observe the cars dataset in the R console, for every instance where speed increases, the distance also increases along with it.

That means, there is a strong positive relationship between them. So, the correlation between them will be closer to 1.

However, correlation doesn't imply causation.

In other words, if two variables have high correlation, it does not mean one variable 'causes' the value of the other variable to increase.

Correlation is only an aid to understand the relationship. You can only rely on logic and business reasoning to make that judgement.

So, how to compute correlation in R?

Simply use the `cor()` function with the two numeric variables as arguments.

```
cor(cars$speed, cars$dist)  # calculate correlation between speed and distance
```

```
## [1] 0.8068949
```

Build the Linear Regression Model

Now that you have seen the linear relationship pictorially in the scatter plot and through correlation, let's try building the linear regression model.

The function used for building linear models is`lm()`.

The`lm()` function takes in two main arguments:

1.Formula 2.Data

The data is typically a data.frame object and the formula is a object of class formula.

But the most common convention is to write out the formula directly as written below.

```
linearMod <- lm(dist ~ speed, data=cars)  # build linear regression model on full data
print(linearMod)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##     -17.579        3.932
```

By building the linear regression model, we have established the relationship between the predictor and response in the form of a mathematical formula.

That is Distance (`dist`) as a function for `speed`.

For the above output, you can notice the Coefficients part having two components: Intercept: -17.579, speed: 3.932.

These are also called the beta coefficients. In other words,

dist = -17.579 + 3.932*speed

Now the linear model is built and you have a formula that you can use to predict the dist value if a corresponding speed is known.

Is this enough to actually use this model? NO!

Because, before using a regression model to make predictions, you need to ensure that it is statistically significant. But How do you ensure this?

Lets begin by printing the summary statistics for linearMod.

```
summary(linearMod)  # model summary
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```