Linear model - Discover the relationship between speed and distance

w.Lu

13/01/2020

## Introduction

This a Linear model sample according to the car data about distance, speed ,name of the car . we run our data through a modelling algorithm. The model that we will be using is the Linear Regression Model, which is helpful when trying to discover the relationship between two variables. These two variables represent the X and Y within the linear equation. The X variable is the predictor variable, also known as the independent variable because it doesn’t depend on other attributes while making predictions. Y is the response variable, also known as the dependent variable because its value depends on the other variables. (We will be keeping this at a high level. If you’d like to discover more about this equation, please feeI free to do your own research.) In our case, these two variables will be \*\* Speed \*\* and \*\* Distance \*\*. We are trying to predict Distance, so it is our dependent/response/Y variable. Speed is our independent/predictor/X variable. To create this model, we will be using the linear model function – lm(). Here is the basic line of code for the linear model function.

# read data  
dataset <- read.csv("cars.csv")  
#List your attributes within your data set.  
attributes(dataset)

## $names  
## [1] "name.of.car" "speed.of.car" "distance.of.car"  
##   
## $class  
## [1] "data.frame"  
##   
## $row.names  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

head(dataset)

## name.of.car speed.of.car distance.of.car  
## 1 Ford 4 2  
## 2 Jeep 4 4  
## 3 Honda 7 10  
## 4 KIA 7 10  
## 5 Toyota 8 14  
## 6 BMW 9 16

#Prints the min, max, mean, median, and quartiles of each attribute.  
summary(dataset)

## name.of.car speed.of.car distance.of.car   
## Dodge : 3 Min. : 4.0 Min. : 2.00   
## Honda : 3 1st Qu.:12.0 1st Qu.: 26.00   
## Jeep : 3 Median :15.0 Median : 36.00   
## KIA : 3 Mean :15.4 Mean : 42.98   
## Acura : 2 3rd Qu.:19.0 3rd Qu.: 56.00   
## Audi : 2 Max. :25.0 Max. :120.00   
## (Other):34

#Displays the structure of your data set.  
str(dataset)

## 'data.frame': 50 obs. of 3 variables:  
## $ name.of.car : Factor w/ 23 levels "Acura","Audi",..: 9 15 12 16 23 3 20 10 13 14 ...  
## $ speed.of.car : int 4 4 7 7 8 9 10 10 10 11 ...  
## $ distance.of.car: int 2 4 10 10 14 16 17 18 20 20 ...

#Names your attributes within your data set.  
names(dataset)

## [1] "name.of.car" "speed.of.car" "distance.of.car"

#Will print out the instances within that particular column in your data set.  
dataset$name.of.car

## [1] Ford Jeep Honda KIA Toyota BMW   
## [7] Mercedes GM Hyundai Infiniti Land Rover Lexus   
## [13] Mazda Mitsubishi Nissan GMC Fiat Chrysler   
## [19] Dodge Acura Audi Chevrolet Buick Ford   
## [25] Jeep Honda KIA Toyota BMW Mercedes   
## [31] GM Hyundai Infiniti Land Rover Lexus Mazda   
## [37] Mitsubishi Nissan GMC Fiat Chrysler Dodge   
## [43] Acura Audi Chevrolet Buick Jeep Honda   
## [49] KIA Dodge   
## 23 Levels: Acura Audi BMW Buick Chevrolet Chrysler Dodge Fiat Ford GM ... Toyota

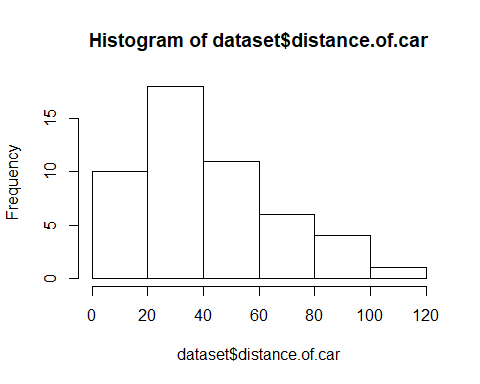
dataset$speed.of.car

## [1] 4 4 7 7 8 9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15  
## [26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24 25

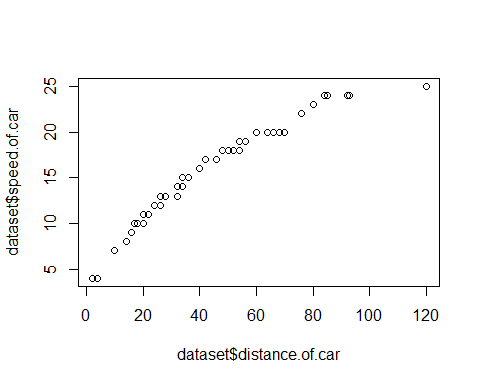
dataset$distance.of.car

## [1] 2 4 10 10 14 16 17 18 20 20 22 24 26 26 26 26 28 28 32  
## [20] 32 32 34 34 34 36 36 40 40 42 46 46 48 50 52 54 54 56 56  
## [39] 60 64 66 68 70 76 80 84 85 92 93 120

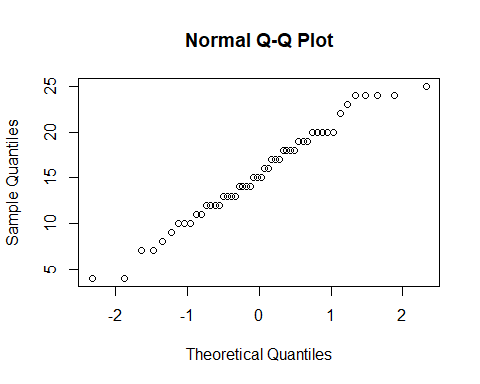
# Your columns must be in numeric form to perform these plots.  
hist(dataset$distance.of.car)



plot(dataset$distance.of.car,dataset$speed.of.car)



qqnorm(dataset$speed.of.car)



# Data Types  
#Numeric- Numbers with decimals. (Ex: 1.0, 10.5, 4.5, etc.)  
#Integer Data- Whole numbers (Ex: 11, 45, 78, etc.)  
#Factor Data- Categorical data (Ex: Red, Blue, Green, Purple, etc.)  
#Ordinal Data- Ordered data (Ex: educational levels, temperature, etc.)  
#Character Data- String values, which are characters (words) with quotes around them. (Ex: "Red", "Blue", "Green", "Purple", etc.)  
#Logical- TRUE or TRUE (Always capitalize TRUE or FALSE)  
  
  
names(dataset)<-c("Brand","Speed","Distance")   
summary(dataset)

## Brand Speed Distance   
## Dodge : 3 Min. : 4.0 Min. : 2.00   
## Honda : 3 1st Qu.:12.0 1st Qu.: 26.00   
## Jeep : 3 Median :15.0 Median : 36.00   
## KIA : 3 Mean :15.4 Mean : 42.98   
## Acura : 2 3rd Qu.:19.0 3rd Qu.: 56.00   
## Audi : 2 Max. :25.0 Max. :120.00   
## (Other):34

#Will show your NA’s through logical data. (TRUE if it’s missing, FALSE if it’s not.)  
is.na(dataset)

## Brand Speed Distance  
## [1,] FALSE FALSE FALSE  
## [2,] FALSE FALSE FALSE  
## [3,] FALSE FALSE FALSE  
## [4,] FALSE FALSE FALSE  
## [5,] FALSE FALSE FALSE  
## [6,] FALSE FALSE FALSE  
## [7,] FALSE FALSE FALSE  
## [8,] FALSE FALSE FALSE  
## [9,] FALSE FALSE FALSE  
## [10,] FALSE FALSE FALSE  
## [11,] FALSE FALSE FALSE  
## [12,] FALSE FALSE FALSE  
## [13,] FALSE FALSE FALSE  
## [14,] FALSE FALSE FALSE  
## [15,] FALSE FALSE FALSE  
## [16,] FALSE FALSE FALSE  
## [17,] FALSE FALSE FALSE  
## [18,] FALSE FALSE FALSE  
## [19,] FALSE FALSE FALSE  
## [20,] FALSE FALSE FALSE  
## [21,] FALSE FALSE FALSE  
## [22,] FALSE FALSE FALSE  
## [23,] FALSE FALSE FALSE  
## [24,] FALSE FALSE FALSE  
## [25,] FALSE FALSE FALSE  
## [26,] FALSE FALSE FALSE  
## [27,] FALSE FALSE FALSE  
## [28,] FALSE FALSE FALSE  
## [29,] FALSE FALSE FALSE  
## [30,] FALSE FALSE FALSE  
## [31,] FALSE FALSE FALSE  
## [32,] FALSE FALSE FALSE  
## [33,] FALSE FALSE FALSE  
## [34,] FALSE FALSE FALSE  
## [35,] FALSE FALSE FALSE  
## [36,] FALSE FALSE FALSE  
## [37,] FALSE FALSE FALSE  
## [38,] FALSE FALSE FALSE  
## [39,] FALSE FALSE FALSE  
## [40,] FALSE FALSE FALSE  
## [41,] FALSE FALSE FALSE  
## [42,] FALSE FALSE FALSE  
## [43,] FALSE FALSE FALSE  
## [44,] FALSE FALSE FALSE  
## [45,] FALSE FALSE FALSE  
## [46,] FALSE FALSE FALSE  
## [47,] FALSE FALSE FALSE  
## [48,] FALSE FALSE FALSE  
## [49,] FALSE FALSE FALSE  
## [50,] FALSE FALSE FALSE

set.seed(10)  
trainSize<- round(nrow(dataset) \* 0.7)  
testSize <- nrow(dataset) - trainSize  
  
training\_indices <- sample(seq\_len(nrow(dataset)), size = trainSize)  
trainSet <- dataset[training\_indices,]  
testSet <- dataset[-training\_indices,]   
ModelA <- lm(dataset$Speed~dataset$Distance, trainSet)  
summary(ModelA)

##   
## Call:  
## lm(formula = dataset$Speed ~ dataset$Distance, data = trainSet)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.6279 -0.3259 0.3640 0.9800 1.7938   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.902268 0.395965 17.43 <2e-16 \*\*\*  
## dataset$Distance 0.197714 0.007922 24.96 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.429 on 48 degrees of freedom  
## Multiple R-squared: 0.9284, Adjusted R-squared: 0.927   
## F-statistic: 622.8 on 1 and 48 DF, p-value: < 2.2e-16

# Multiple R-squared- How well the regression line fits the data (1 means it’s a perfect fit).  
# A p-value of more than 0.05 means the Independent Variable has no effect on the Dependent Variable; less than 0.05 means the relationship is statistically significant.  
Predictiondistance <- predict(ModelA,trainSet)

## Warning: 'newdata' had 35 rows but variables found have 50 rows

Predictiondistance

## 1 2 3 4 5 6 7 8   
## 7.297695 7.693123 8.879404 8.879404 9.670259 10.065686 10.263400 10.461113   
## 9 10 11 12 13 14 15 16   
## 10.856541 10.856541 11.251968 11.647395 12.042822 12.042822 12.042822 12.042822   
## 17 18 19 20 21 22 23 24   
## 12.438250 12.438250 13.229104 13.229104 13.229104 13.624532 13.624532 13.624532   
## 25 26 27 28 29 30 31 32   
## 14.019959 14.019959 14.810813 14.810813 15.206241 15.997095 15.997095 16.392522   
## 33 34 35 36 37 38 39 40   
## 16.787950 17.183377 17.578804 17.578804 17.974232 17.974232 18.765086 19.555941   
## 41 42 43 44 45 46 47 48   
## 19.951368 20.346795 20.742222 21.928504 22.719359 23.510213 23.707927 25.091922   
## 49 50   
## 25.289636 30.627904