2

February 1, 2024

1 Linear Regression In R

- load the car data using data() and see what it contains

[45]:
```
# Load the 'cars' dataset
data(cars)

# View the dataset
View(cars)
```

A data.frame: 50 × 2

speed	dist
<dbl>	<dbl>
4	2
4	10
7	4
7	22
8	16
9	10
10	18
10	26
10	34
11	17
11	28
12	14
12	20
12	24
12	28
13	26
13	34
13	34
13	46
14	26
14	36
14	60
14	80
15	20
15	26
15	54
16	32
16	40
17	32
17	40
17	50
18	42
18	56
18	76
18	84
19	36
19	46
19	68
20	32
20	48
20	52
20	56
20	64
22	66
23	54
24	70
24	92
24	93
24	120
25	85

2 Visualization and Training the Model

[46]:
```
install.packages('caTools')
library(caTools)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

we will train a model of the form $Y = \ 1 + \ 2X + \ $ where Y is the car breaking distance and X is the car's speed. To train a linear model on the data, we use the lm() command.

[47]:
```
split = sample.split(cars$dist, SplitRatio = 0.7)
trainingset = subset(cars, split == TRUE)
testset = subset(cars, split == FALSE)

# Fitting Simple Linear Regression to the Training set
model = lm(formula = cars$dist ~ cars$speed)
```

[48]:
```
# Display a summary of the linear regression model
summary(model)
```

```

Call:
lm(formula = cars$dist ~ cars$speed)

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 *
cars$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
```

We now have a trained linear model that predicts the stopping distance of a car given its speed.

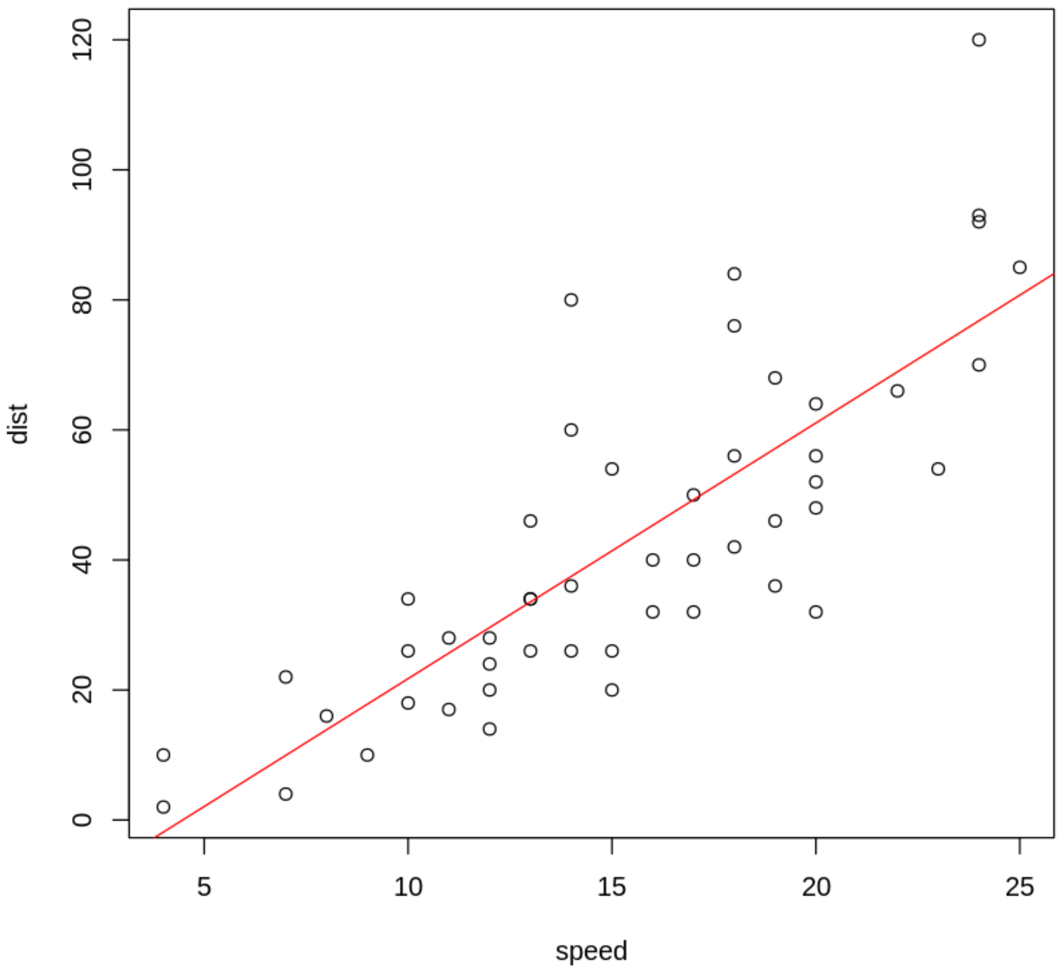
Regression Line : Distance = -17.58+3.93*Speed

To visualize our regression line, we can overlay it with the original training data.

[49]:

```r
# Create a scatter plot of 'dist' against 'speed'
plot(dist ~ speed, data = cars)

# Add the regression line to the plot in red color
abline(model, col = "red")
```



To view additional details of the model, use the summary() command:

summary provides lots of data on the model such as the R squared and adjusted R squared values, the F statistic, and the p-value and is a valuable tool for evaluating the model. Since $p < 0.05$, we can reject the null hypothesis.

We can also view other information such as the error sum of squares and mean sum of squares through the anova() command.

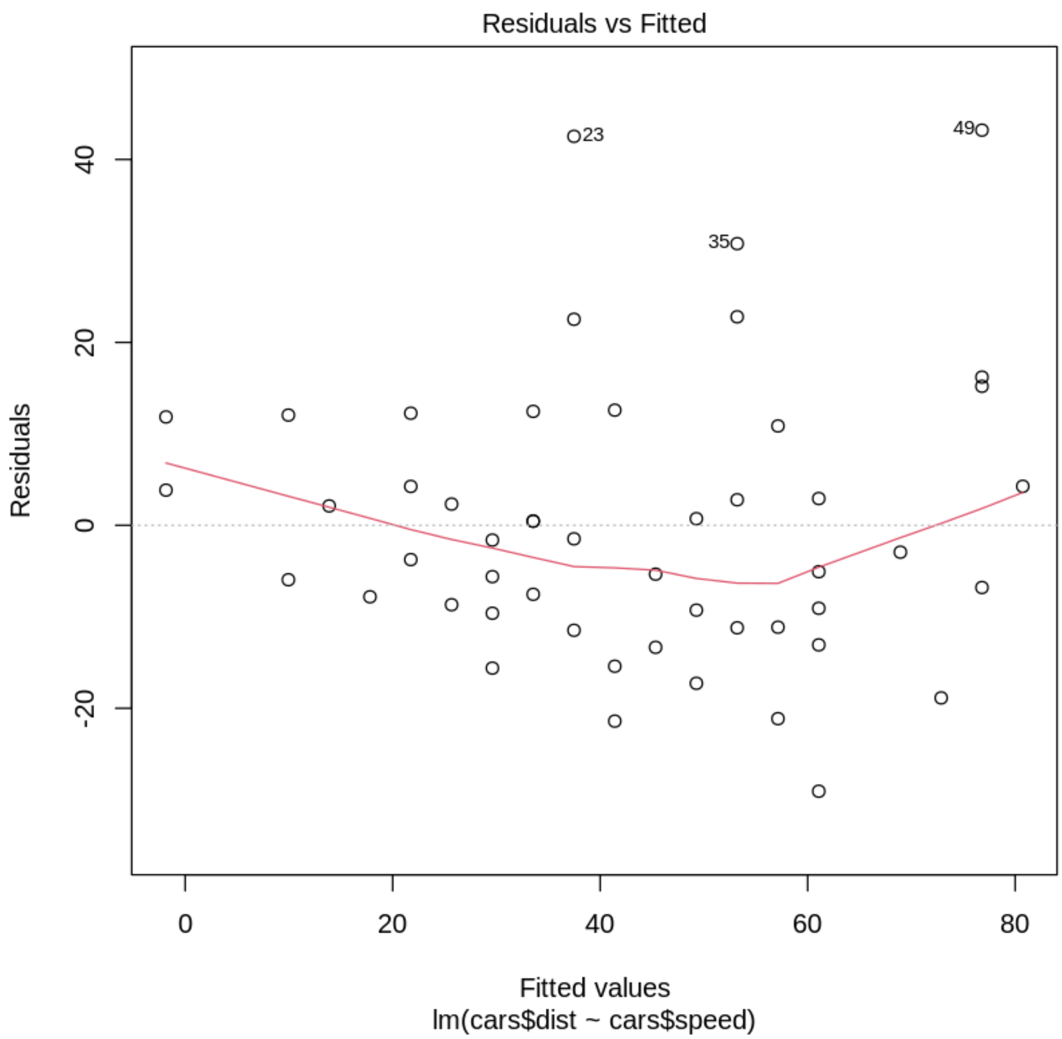
[50]:
```
# Perform an analysis of variance (ANOVA) on the linear regression model
anova(model)
```

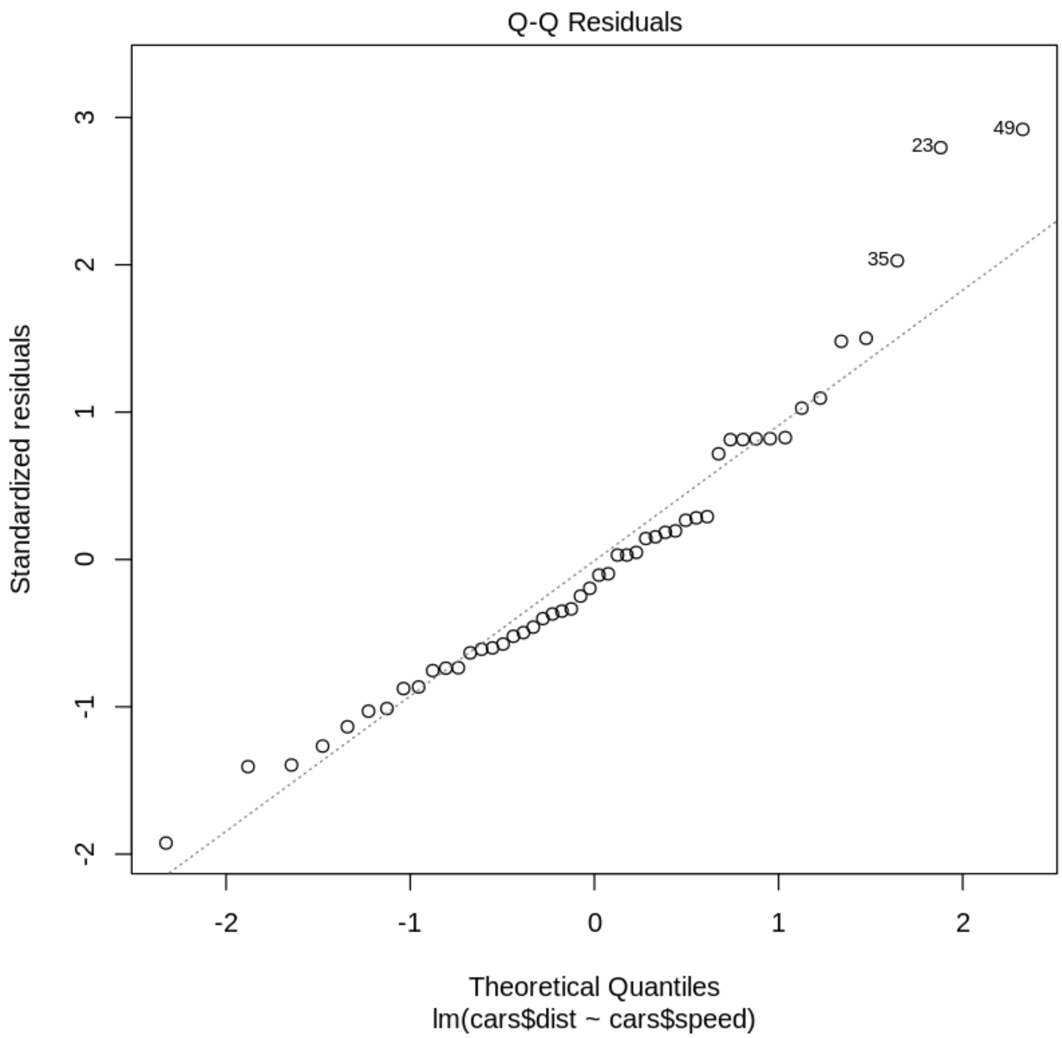
A anova: 2 × 5

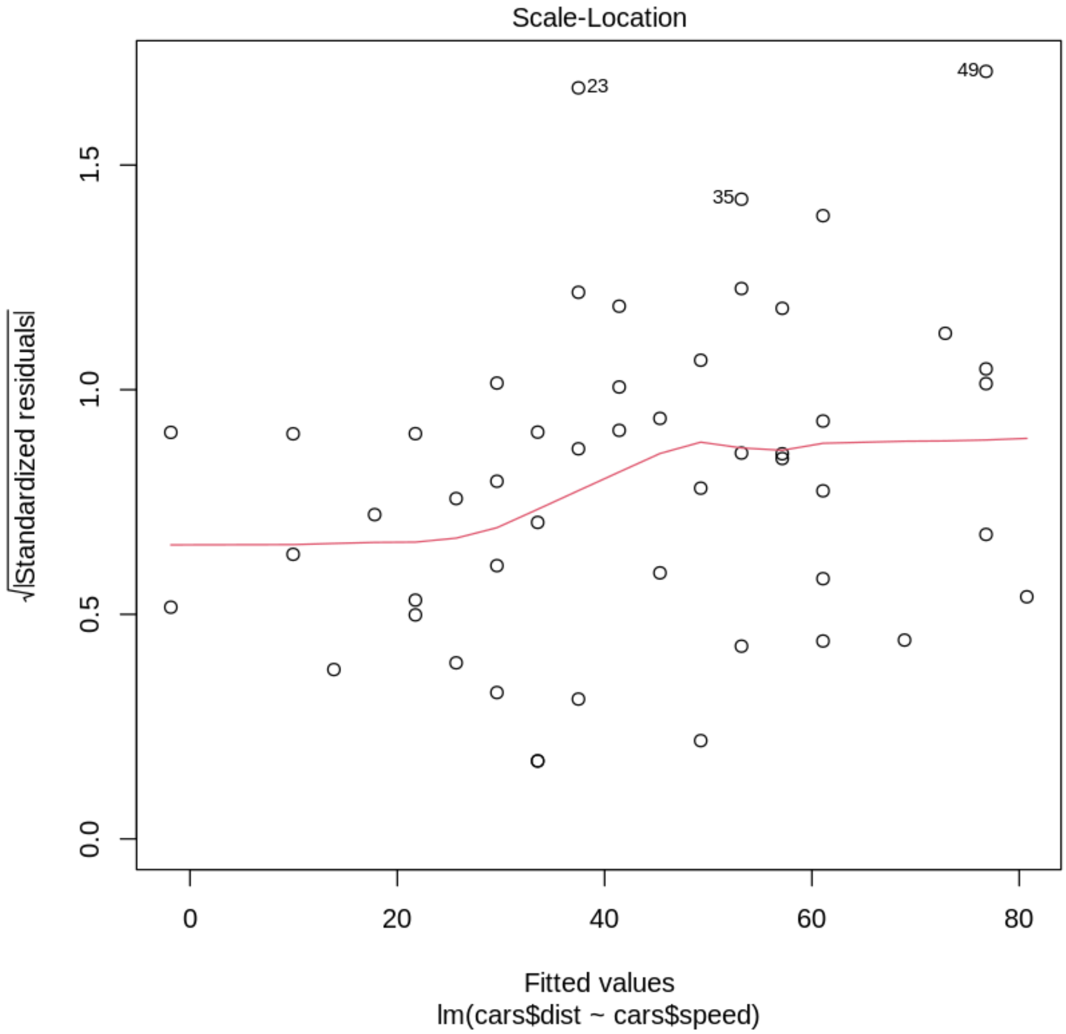
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
cars$speed	1	21185.46	21185.4589	89.56711	1.489836e-12
Residuals	48	11353.52	236.5317	NA	NA

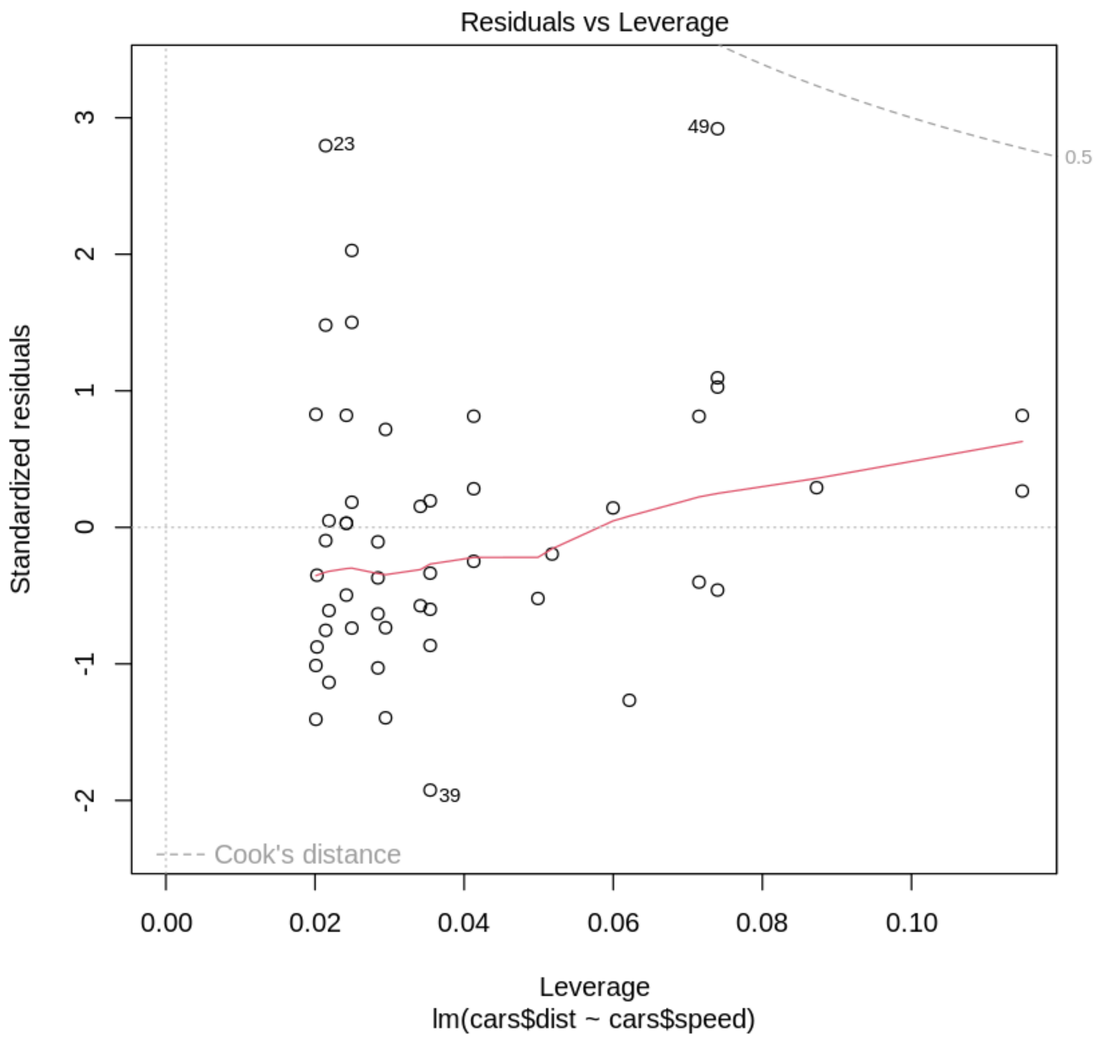
Additionally, we can graphically analyze the statistical properties of our model.

[51]:
```
# Plot diagnostic plots for the linear regression model
plot(model)
```




Q-Q Residuals
Standardized residuals
3
2
1
0
-1
-2
49
23
35
-2
-1
0
1
2
Theoretical Quantiles
lm(cars$dist ~ cars$speed)


Scale-Location
23
49
35
1.5
1.0
0.5
0.0
√|Standardized residuals|
0
20
40
60
80
Fitted values
lm(cars$dist ~ cars$speed)


Residuals vs Leverage
Standardized residuals
3
2
1
0
-1
-2
0.00
0.02
0.04
0.06
0.08
0.10
Leverage
lm(cars$dist ~ cars$speed)
23
49
39
0.5
Cook's distance