Residual Analysis

Rick Scavetta

26 November 2016

[http://datascienceplus.com/how-to-detect-heteroscedasticity-and-rectify-it/]

Intro

One of the important assumptions of linear regression is that, there should be no heteroscedasticity of residuals. In simpler terms, this means that the variance of residuals should not increase with fitted values of response variable. In this post, I am going to explain why it is important to check for heteroscedasticity, how to detect it in your model? If is present, how to make amends to rectify the problem, with example R codes. This process is sometimes referred to as *residual analysis*.

Why?

It is customary to check for heteroscedasticity of residuals once you build the linear regression model. The reason is, we want to check if the model thus built is unable to explain some pattern in the response variable Y, that eventually shows up in the residuals. This would result in an inefficient and unstable regression model that could yield bizarre predictions later on.

How?

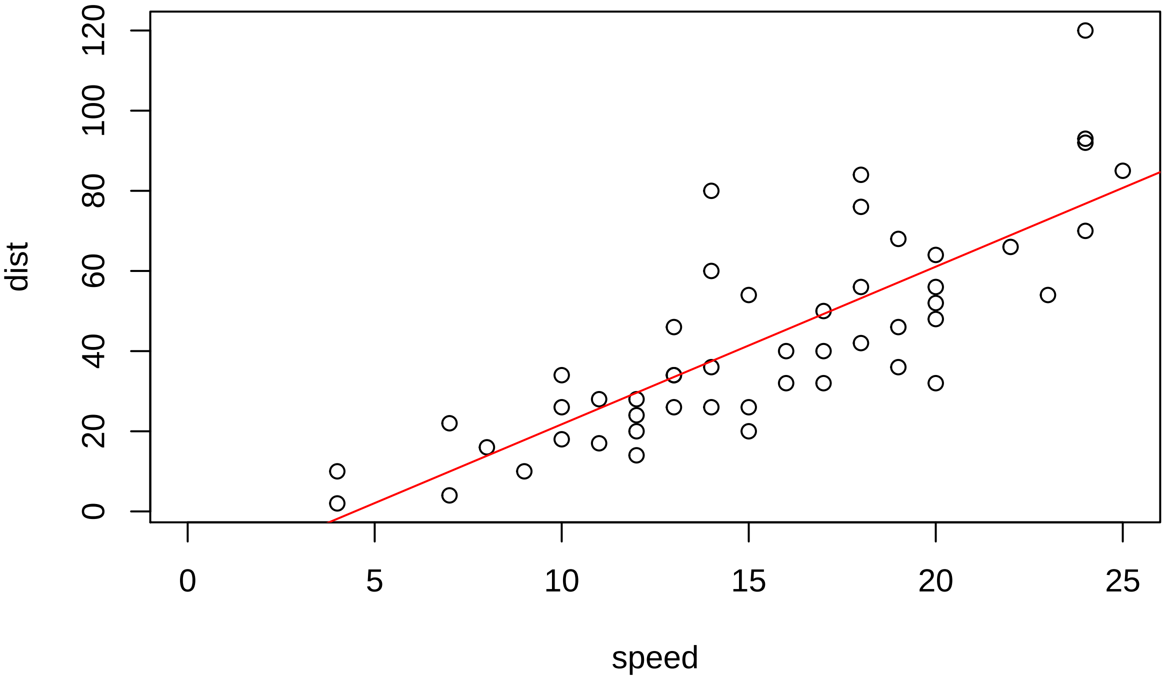
I am going to illustrate this with an actual regression model based on the cars dataset, that comes built-in with R. Lets first build the model using the lm() function.

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

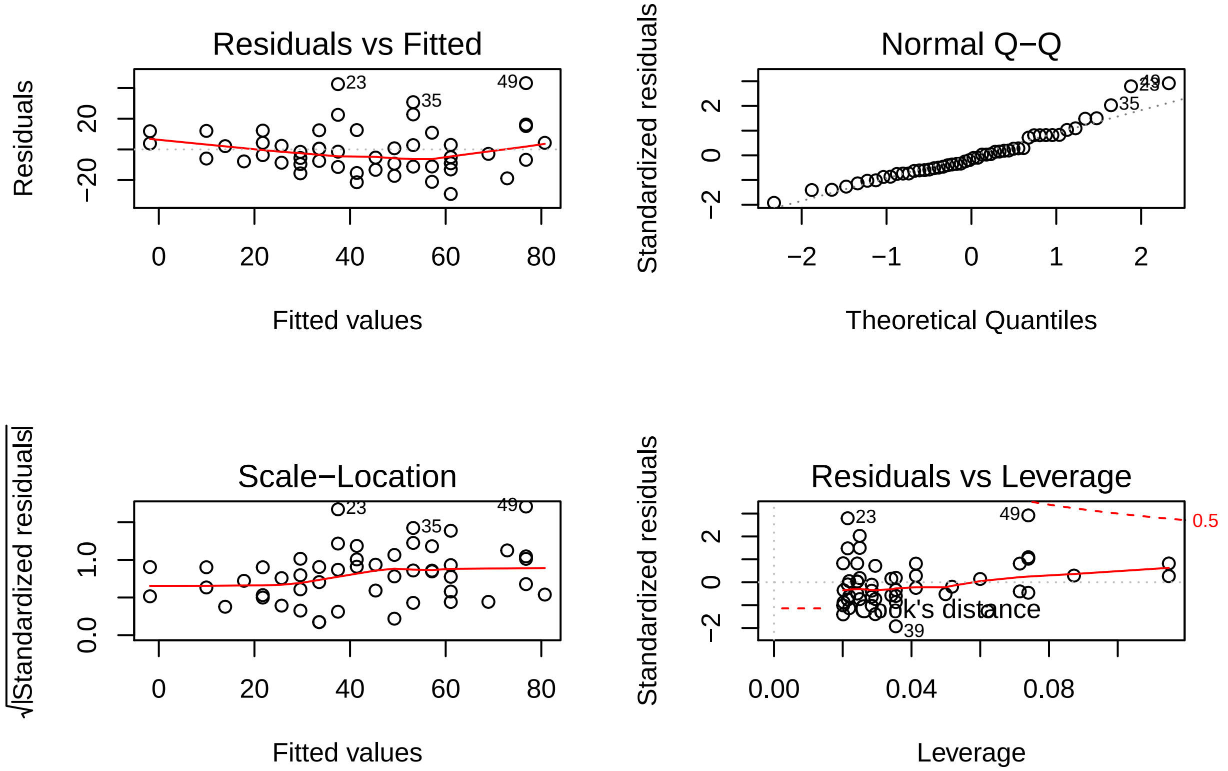
```r
# data
plot(cars, xlim = c(0,25))

# model
lmMod <- lm(dist ~ speed, data=cars) # initial model
abline(lmMod, col = "red")
```



Now that the model is ready, there are two ways to test for heterosedasticity:

1. Graphically



The plots we are interested in are at the top-left and bottom-left.

The top-left is residuals vs fitted values, The bottom-left is standardised residuals on Y axis.

If there is absolutely no heteroscedastity, you should see a completely random, equal distribution of points throughout the range of X axis and a flat red line. But in our case, as you can notice from the top-left plot, the red line is slightly curved and the residuals seem to increase as the fitted Y values increase. So, the inference here is, heteroscedasticity exists.

2. Statistical tests

Sometimes you may want an algorithmic approach to check for heteroscedasticity so that you can quantify its presence automatically and make amends. For this purpose, there are a couple of tests that comes handy to establish the presence or absence of heteroscedasticity – The *Breush-Pagan test* and the *NCV test*.

```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(lmMod)  # Breusch-Pagan test
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmMod
## BP = 3.2149, df = 1, p-value = 0.07297
```

```
library(car)
ncvTest(lmMod)  # NCV Test
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 4.650233    Df = 1     p = 0.03104933
```

Both these test have a p-value less that a significance level of `0.05`, therefore we can reject the null hypothesis that the variance of the residuals is constant and infer that heteroscedasticity is indeed present, thereby confirming our graphical inference.

Fix it

1. Re-build the model with new predictors.

Since we have no other predictors apart from "speed", I can't show this method now. However, one option I might consider trying out is to add the residuals of the original model as a predictor and rebuild the regression model. With a model that includes residuals (as X) whose future actual values are unknown, you might ask what will be the value of the new predictor (i.e. residual) to use on the test data?. The solutions is, for starters, you could use the mean value of residuals for all observations in test data. Though is this not recommended, it is an approach you could try out if all available options fail.

2. Variable transformation such as Box-Cox transformation.

Box-cox transformation is a mathematical transformation of the variable to make it approximate to a normal distribution. Often, doing a box-cox transformation of the Y variable solves the issue.

```
distBCMod <- caret::BoxCoxTrans(cars$dist)
print(distBCMod)
```

```
## Box-Cox Transformation
##
## 50 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   26.00   36.00   42.98   56.00  120.00
##
## Largest/Smallest: 60
## Sample Skewness: 0.759
##
## Estimated Lambda: 0.5
```

The model for creating the box-cox transformed variable is ready. Lets now apply it to `car$dist` and append it to a new dataframe.

```
cars$dist_new <- predict(distBCMod, cars$dist) # append the transformed variable to cars
head(cars) # view the top 6 rows
```

```
##   speed dist  dist_new
## 1     4    2 0.8284271
## 2     4   10 4.3245553
## 3     7    4 2.0000000
## 4     7   22 7.3808315
## 5     8   16 6.0000000
## 6     9   10 4.3245553
```

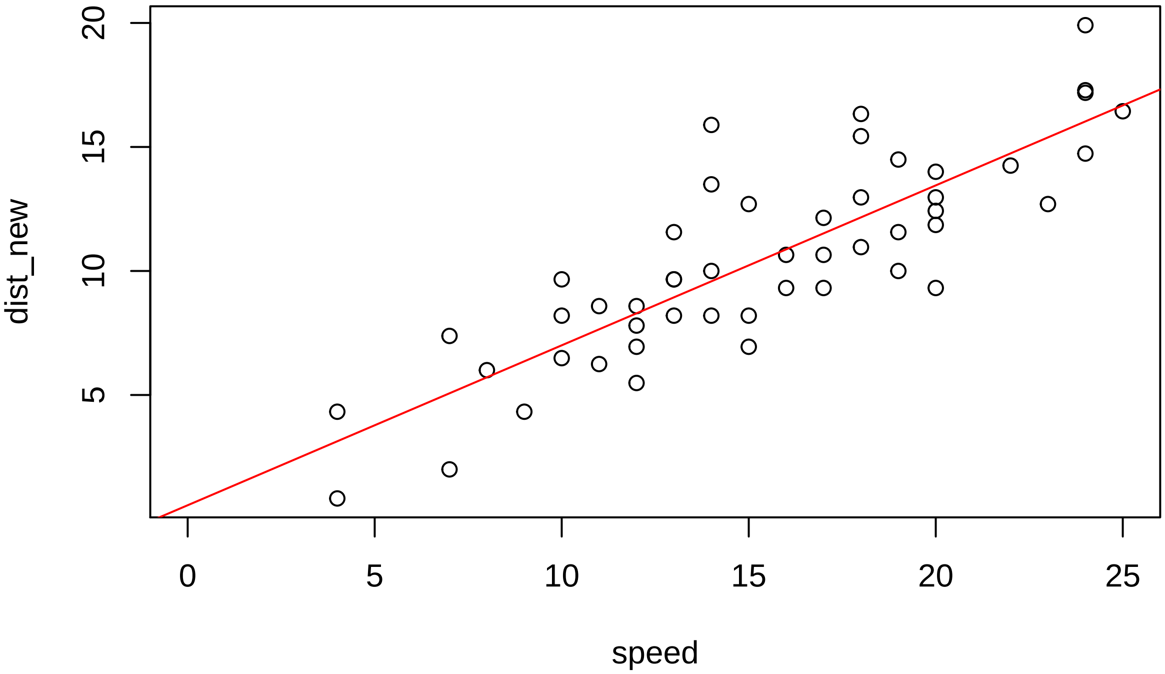
Build the model and check for heteroscedasticity.

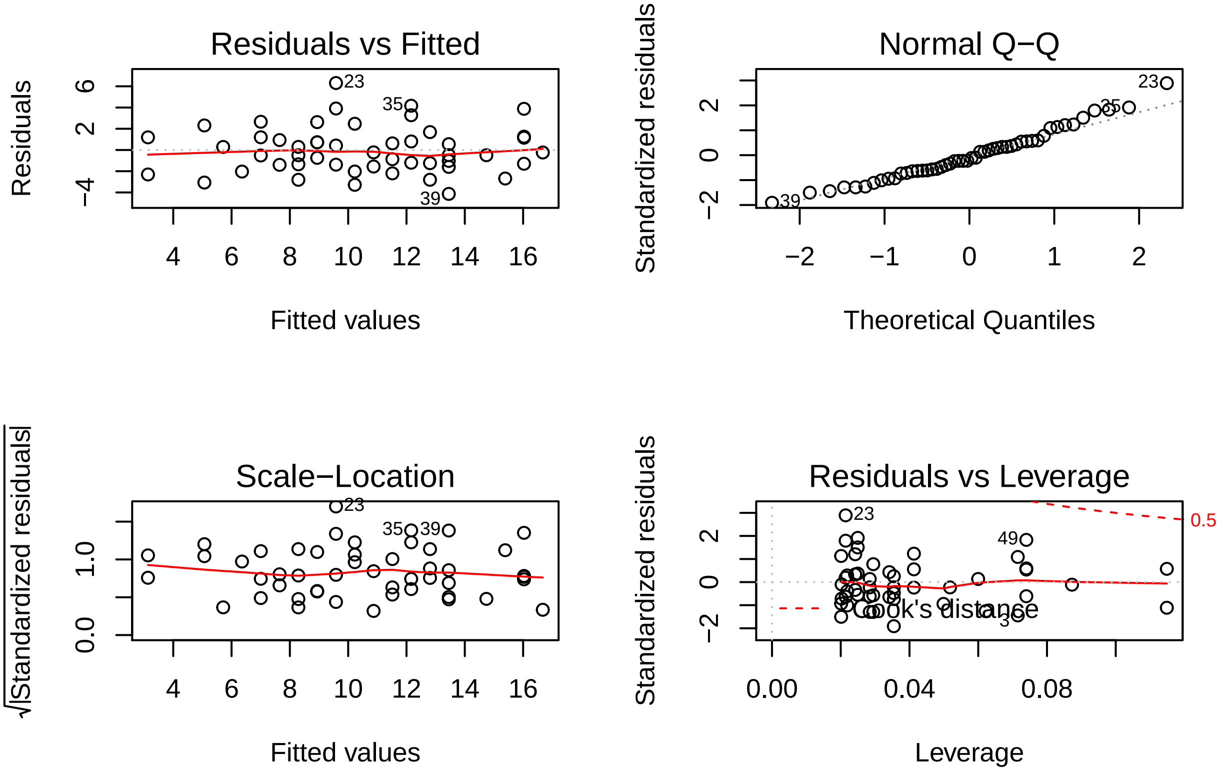
```
lmMod_bc <- lm(dist_new ~ speed, data=cars)
bptest(lmMod_bc)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmMod_bc
## BP = 0.011192, df = 1, p-value = 0.9157
```

With a p-value of 0.91, we fail to reject the null hypothesis (that variance of residuals is constant) and therefore infer that ther residuals are homoscedastic. Lets check this graphically as well.

```
plot(dist_new ~ speed, data=cars, xlim = c(0,25))
abline(lmMod_bc, col = "red")
```


dist_new
5
10
15
20
0
5
10
15
20
25
speed


Residuals vs Fitted
Residuals
−4
2
6
4
6
8
10
12
14
16
Fitted values
23
35
39
Normal Q−Q
Standardized residuals
−2
0
2
−2
−1
0
1
2
Theoretical Quantiles
23
35
39
Scale−Location
√|Standardized residuals|
0.0
1.0
4
6
8
10
12
14
16
Fitted values
23
35
39
Residuals vs Leverage
Standardized residuals
−2
0
2
0.00
0.04
0.08
Leverage
23
49
36
0.5
Cook's distance