Checking Linear Regression Asssumptions in R

A. Linear Regression Overview

There are two types of supervised machine learning algorithms: Regression and classification. The former predicts continuous value outputs while the latter predicts discrete outputs. For instance, predicting the price of a house in dollars is a regression problem whereas predicting whether a tumor is malignant or benign is a classification problem.

This notebook explains the assumptions of linear regression in detail. One of the most essential steps to take before applying linear regression and depending solely on accuracy scores is to check for these assumptions. If any of these assumptions is violated, then the forecasts, confidence intervals, and scientific insights yielded by the regression model may be (at best) inefficient or (at worst) seriously biased or misleading.

```r
# Load Packages

library(ggplot2)
library(GGally)
library(stargazer)
library(car)
```

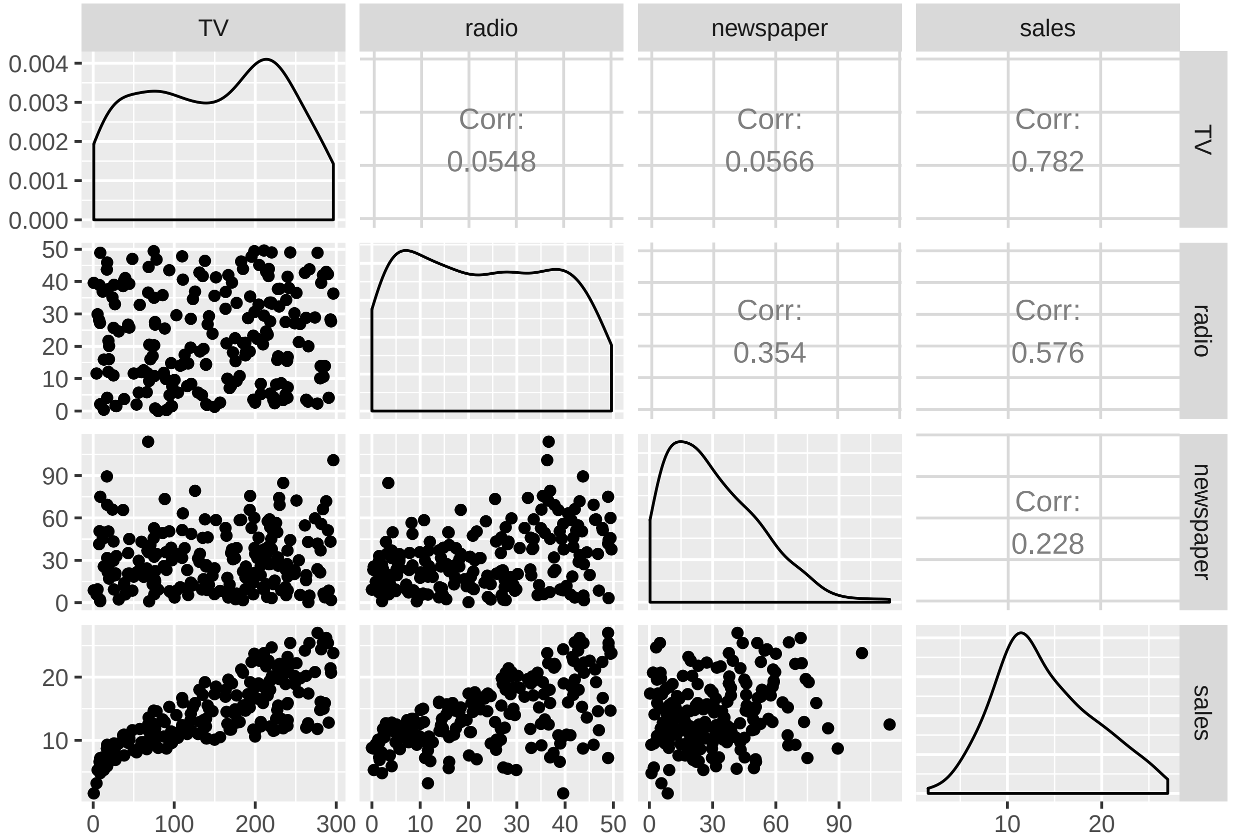
```r
# Load dataset - http://www-bcf.usc.edu/~gareth/ISL/data.html

data = read.csv('Advertising.csv')
ad_data=data[ ,-1]
```

Lets do some basic exploratory analysis of the dataset

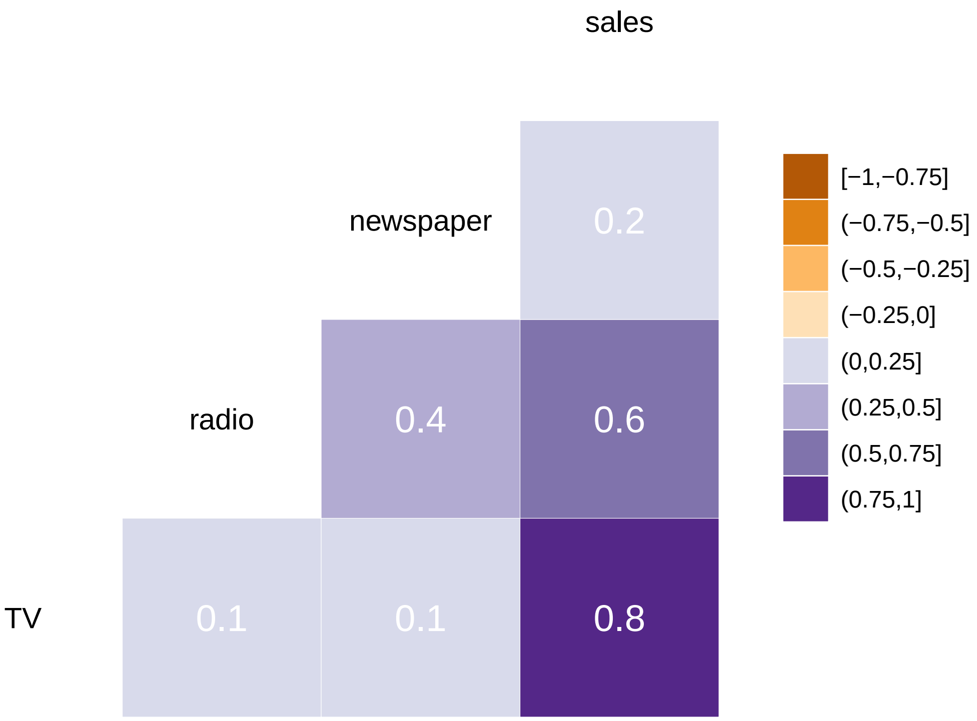
```r
summary(ad_data)
```

```
##        TV             radio          newspaper          sales      
##  Min.   :  0.70   Min.   : 0.000   Min.   :  0.30   Min.   : 1.60  
##  1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75   1st Qu.:10.38  
##  Median :149.75   Median :22.900   Median : 25.75   Median :12.90  
##  Mean   :147.04   Mean   :23.264   Mean   : 30.55   Mean   :14.02  
##  3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10   3rd Qu.:17.40  
##  Max.   :296.40   Max.   :49.600   Max.   :114.00   Max.   :27.00  
```

```r
ggpairs(ad_data)
```



```r
ggcorr(ad_data, nbreaks=8, label=TRUE, label_size=5, label_color='white',palette = "PuOr")
```



Some of the assumptions require us to have performed regression before we can check for them. So let's perform regression on our dataset.

```r
#use lm() function here to run your regression model
regr<-lm(sales~ TV + radio + newspaper, data=ad_data)
```

```r
summary(regr)
```

```
##
## Call:
## lm(formula = sales ~ TV + radio + newspaper, data = ad_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
#Stargazer package gave us a nice summary table

stargazer(regr,type="text")
```

```
## 
## ===============================================
##                         Dependent variable:    
##                     ---------------------------
##                                sales           
## -----------------------------------------------
## TV                           0.046***          
##                               (0.001)          
## 
## radio                        0.189***          
##                               (0.009)          
## 
## newspaper                     -0.001           
##                               (0.006)          
## 
## Constant                     2.939***          
##                               (0.312)          
## 
## -----------------------------------------------
## Observations                    200            
## R2                             0.897           
## Adjusted R2                    0.896           
## Residual Std. Error      1.686 (df = 196)      
## F Statistic          570.271*** (df = 3; 196)  
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

B. Assumptions of Linear Regression

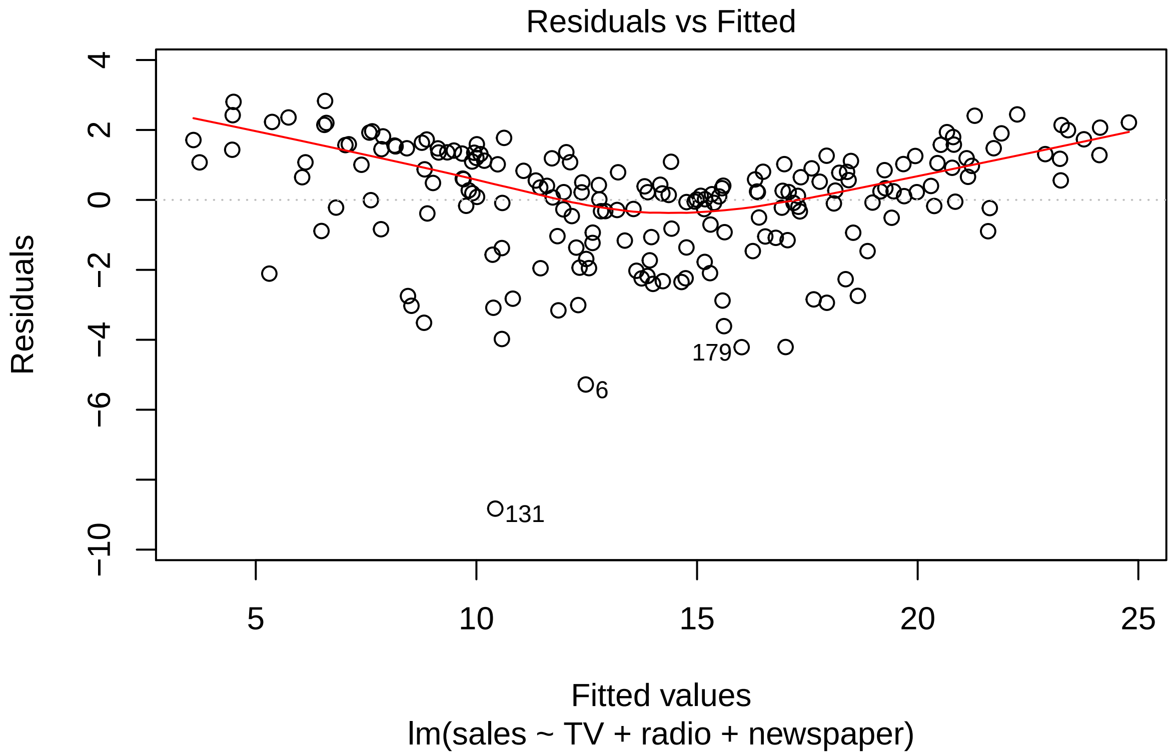
1. The regression model is linear in parameters
2. Homoscedasticity of residuals or equal variance
3. The mean of residuals is zero
4. No autocorrelation of residuals
5. Normality of residuals
6. No perfect multicollinearity

Lets run the plot() function to run the different diagnostic plots. We can visually inspect these plots to get a sense of how we did on several of the assumptions outlined above.

[1] a plot of residuals against fitted values, [2] a Scale-Location plot of sqrt(| residuals |) against fitted values, [3] a Normal Q-Q plot, [4] a plot of Cook's distances versus row labels, [5] a plot of residuals against leverages

The regression model is linear in parameters and Homoscedasticity of residuals or equal variance

```r
#Residuals vs fitted plot

plot(regr, which =1)
```



The first plot depicts residuals versus fitted values. The plot of residuals versus predicted values is useful for checking the assumption of linearity and homoscedasticity. If the model does not meet the linear model assumption, we would see our residuals take on a defined shape or a distinctive pattern. The scatterplot of residuals should look like the night sky-no distinctive patterns. The red line through the scatterplot should also be straight and horizontal, not curved, if the linearity assumption is satisfied. To assess if the homoscedasticity assumption is met we look to make sure that the residuals are equally spread around the y = 0 line.

How did we do? R automatically flagged 3 data points that have large residuals (observations 6, 179, and 131). Also, our residuals appear to be non linear. We might consider applying a transformation (log, polynomial) to the dependent and/or independent variables.

We can also run the Breusch-Pagan test to check for heteroskedasticity. The test fits a linear regression model to the residuals of a linear regression model (by default the same explanatory variables are taken as in the main regression model) and rejects if too much of the variance is explained by the additional explanatory variables.

```r
# Breush Pagan Test

lmtest::bptest(regr)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  regr
## BP = 5.1329, df = 3, p-value = 0.1623
```

The p value is quite large which indicates that our data is homoscedastic

3. The mean of residuals is zero

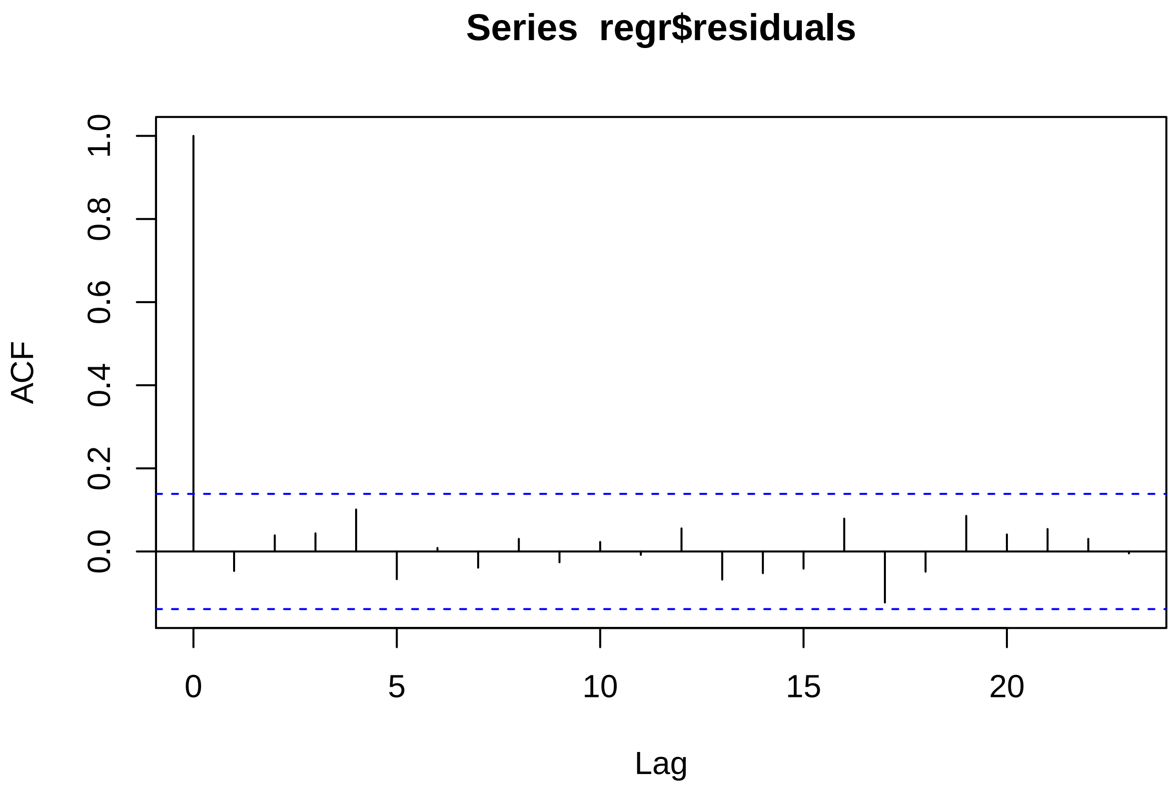
```
mean(regr$residuals)
```

```
## [1] 3.009962e-17
```

Very close to zero so all good here.

4. No autocorrelation of residuals

```
acf(regr$residuals)
```



If the residuals were not autocorrelated, the correlation (Y-axis) from the immediate next line onwards will drop to a near zero value below the dashed blue line (significance level). Clearly, this is not the case here. So we can conclude that the residuals are autocorrelated.

Next lets calculate the Durbin-Watson statistic which is a test statistic used to detect the presence of autocorrelation at lag 1 in the residuals (prediction errors) from a regression analysis.

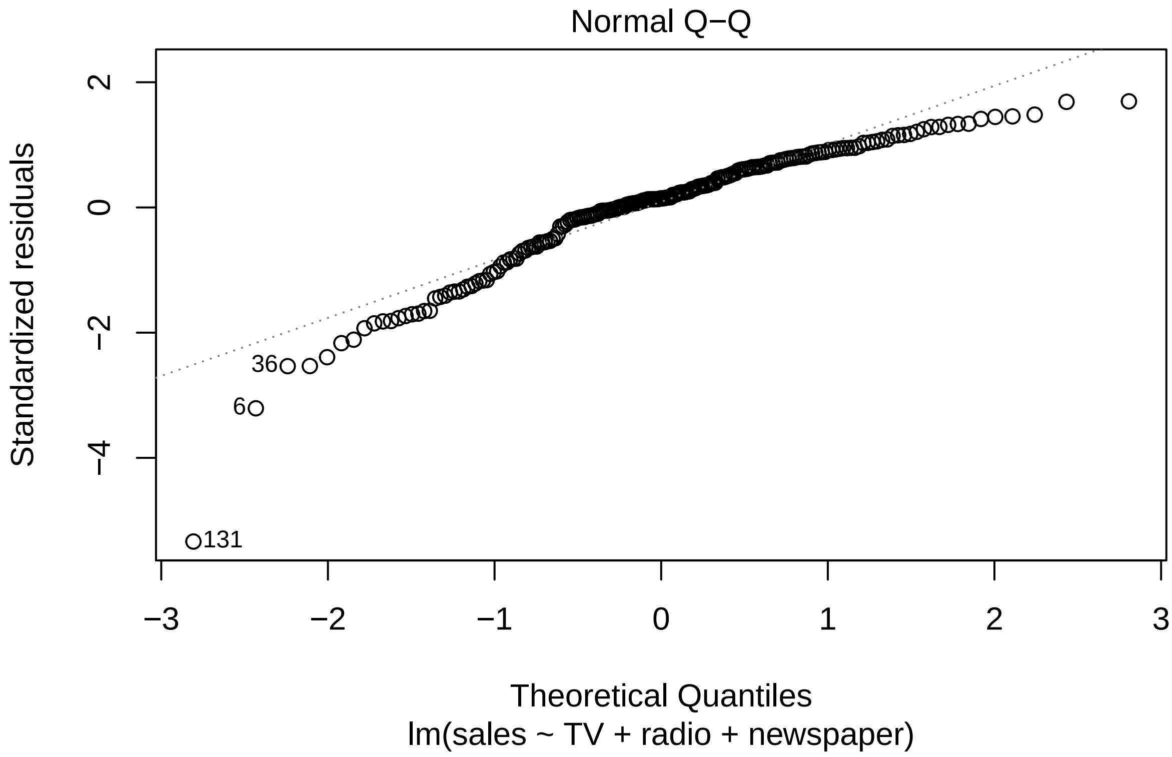
```
# Durbin-Watson

lmtest::dwtest(regr)
```

```
##
##  Durbin-Watson test
##
## data:  regr
## DW = 2.0836, p-value = 0.7236
## alternative hypothesis: true autocorrelation is greater than 0
```

No autocorrelation

5. Normality of residuals

```
plot(regr, which =2)
```



The normality assumption is evaluated based on the standardized residuals and can be evaluated using a QQ-plot by comparing the residuals to "ideal" normal observations along the dotted line. Standardised residuals, also known as internally studentised residuals, are the residuals divided by their estimated standard errors. They are used to adjust for the fact that different residuals have different variances.

R automatically flagged those same 3 data points that have large residuals (observations 116, 187, and 202). Also, there appears to be significant deviation from the ideal normal line. Let us test this statistically.

QQ plot

Q-Q plots take your sample data, sort it in ascending order, and then plot them versus quantiles calculated from a theoretical distribution. The number of quantiles is selected to match the size of your sample data.

- https://stats.stackexchange.com/questions/52212/qq-plot-does-not-match-histogram/52221#52221
- https://stats.stackexchange.com/questions/101274/how-to-interpret-a-qq-plot/101290#101290

```
shapiro.test(regr$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  regr$residuals
## W = 0.91767, p-value = 3.939e-09
```

The shapiro.test tests the NULL hypothesis that the samples came from a Normal distribution. This means that if your p-value <= 0.05, then you would reject the NULL hypothesis that the samples came from a Normal distribution.

6. No perfect multicollinearity

```
vif(regr)
```

```
##        TV     radio newspaper 
##  1.004611  1.144952  1.145187
```

Using Variance Inflation factor (VIF). But, What is VIF?

VIF is a metric computed for every X variable that goes into a linear model. If the VIF of a variable is high, it means the information in that variable is already explained by other X variables present in the given model, which means, more redundant is that variable. So, lower the VIF (<2) the better. VIF for a X var is calculated as:

VIF=1/(1-Rsq)

where, Rsq is the Rsq term for the model with given X as response against all other Xs that went into the model as predictors.

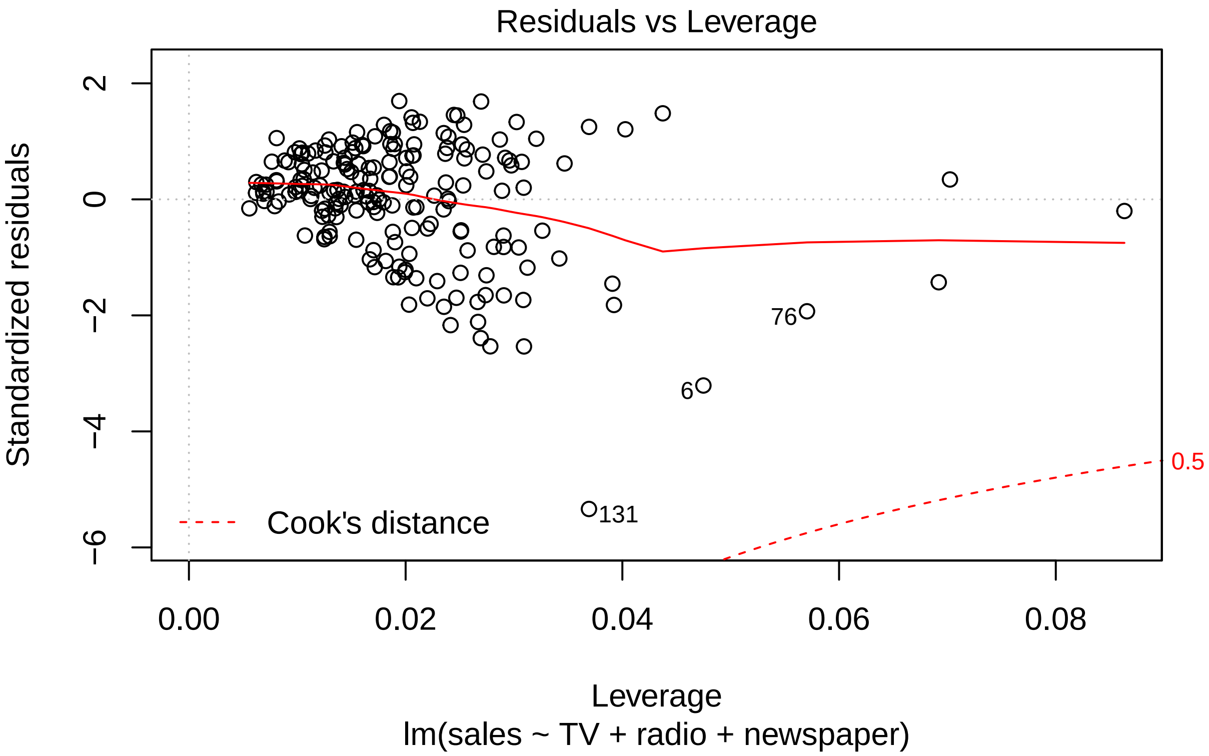
Practically, if two of the X's have high correlation, they will likely have high VIFs. Generally, VIF for an X variable should be less than 4 in order to be accepted as not causing multi-collinearity. The cutoff is kept as low as 2, if you want to be strict about your X variables.

How to rectify? Two ways:

Either iteratively remove the X var with the highest VIF or, See correlation between all variables and keep only one of all highly correlated pairs.

Residuals vs leverage plot

```
plot(regr, which =5)
```



To understand leverage, recognize that Ordinary Least Squares regression fits a line that will pass through the center of your data, $(\bar{X}, \bar{Y})$. The line can be shallowly or steeply sloped, but it will pivot around that point like a lever on a fulcrum. We can take this analogy fairly literally: because OLS seeks to minimize the vertical distances between the data and the line*, the data points that are further out towards the extremes of X will push / pull harder on the lever (i.e., the regression line); they have more *leverage*. One result of this *could* be that the results you get are driven by a few data points; that's what this plot is intended to help you determine.

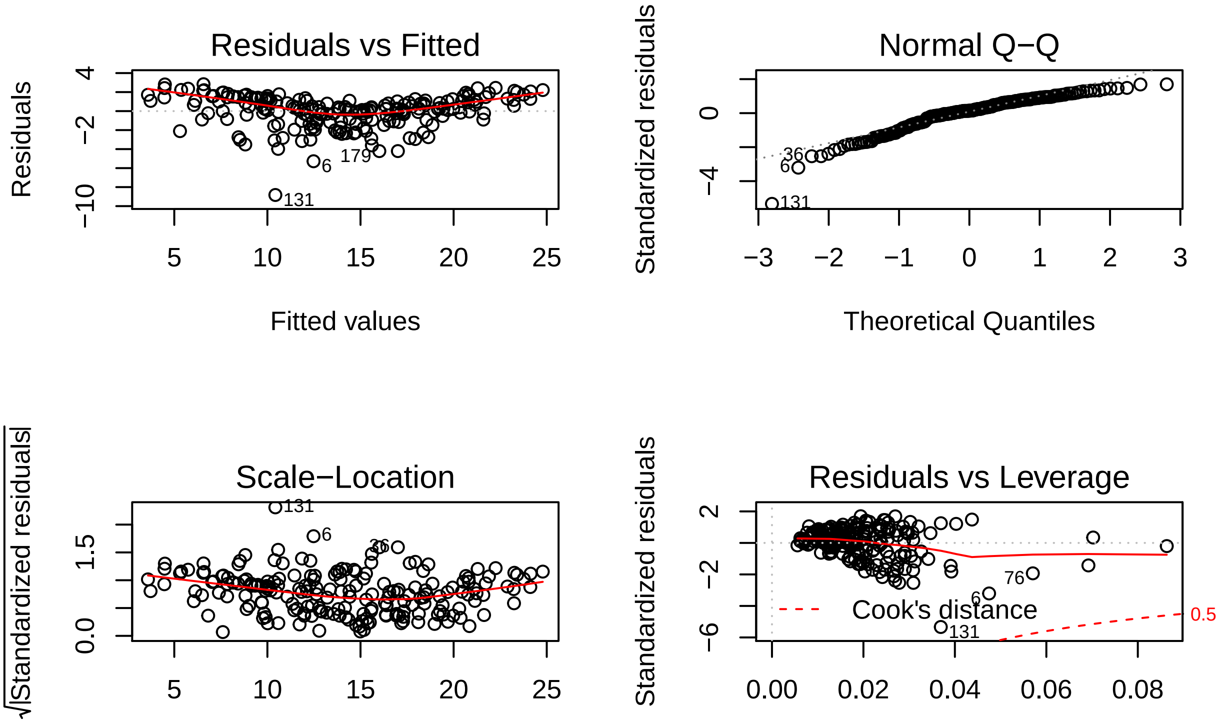
Another result of the fact that points further out on X have more leverage is that they tend to be closer to the regression line (or more accurately: the regression line is fit so as to be closer to *them*) than points that are near $\bar{X}$. In other words, the *residual* standard deviation can differ at different points on X (even if the *error* standard deviation is constant). To correct for this, residuals are often standardized so that they have constant variance (assuming the underlying data generating process is homoscedastic, of course).

One way to think about whether or not the results you have were driven by a given data point is to calculate how far the predicted values for your data would move if your model were fit *without* the data point in question. This calculated total distance is called Cook's distance. Fortunately, you don't have to rerun your regression model N times to find out how far the predicted values will move, Cook's D is a function of the leverage and standardized residual associated with each data point.

Cook's distance or Cook's D is a commonly used estimate of the influence of a data point when performing a least-squares regression analysis. In a practical ordinary least squares analysis, Cook's distance can be used in several ways: to indicate influential data points that are particularly worth checking for validity; or to indicate regions of the design space where it would be good to be able to obtain more data points. Cook's distanceis defined as the sum of all the changes in the regression model when an observation is removed from it.

If you're good and don't have influential cases, you will hardly see a dash red curve, if at all (that red dashed curved line represents Cook's distance). If you don't see a red Cook's distance curved line, or one is just barely peaking out of the corner or your plot but none of your data points are within it, you're good. If you notice some of your data points cross that distance line, you're not so good/ you have influential data points.

```
par(mfrow=c(2,2)) # init 4 charts in 1 panel
plot(regr)
```



Few other things that are important to keep in mind are, the regression model is correctly specified, the number of observations must be greater than number of Xs,the X variables and residuals are uncorrelated and the variability in X values is positive

** References **

- http://r-statistics.co/Assumptions-of-Linear-Regression.html
- https://www.statisticssolutions.com/assumptions-of-linear-regression/
- http://people.duke.edu/~rnau/testing.htm
- https://robert-alvarez.github.io/2018-06-04-diagnostic_plots/ (Great blog for R to Python converts)
- https://en.wikipedia.org/wiki/Cook%27s_distance
- http://strata.uga.edu/8370/rtips/regressionPlots.html