Chapter 3: Linear Regression

Esben Eickhardt

2023-06-20

Introduction

In this chapter we focus on *linear regressions*, how they work and what kind of problems they can solve.

All the same things can also be learned by following the following Statquest playlist.

What Questions can we Answer with Linar Regressions?

Suppose we have a data set with sales as well as budgets for various types of advertisement, what kind of questions would we be able to answer:

- Is there a relationship between advertising budget and sales?
- How strong is the relationship between advertising budget and sales?
- Which media are associated with sales?
- How large is the association between each medium and sales?
- How accurately can we predict future sales?
- Is the relationship linear?
- Is there synergy among the advertising media?

1. Simple Linear Regression

Mathematically we can write a *simple linear regression* as:

$Y = \beta_0 + \beta_1 X$

This could e.g. be a relationship between a TV-advertisement budget and sales, where β_0 and β_1 are two unknown *coefficients* that represent the *intercept* with the y-axis and the *slope* of the line.

$Sales = \beta_0 + \beta_1 * TV$

1.1 Estimating the Coefficients

Our aim is to fit a linear line through the data that minimizes the distances between the predicted values and the actual values. To estimate the *coefficients* for such a line we use data.

The distances between a predicted values and an actual value is called a *residual*, and *residuals* for all the data points can be seen in **Figure 1** as grey lines.

A residual is calculated as the difference between the actual value and the predicted value in the following way:

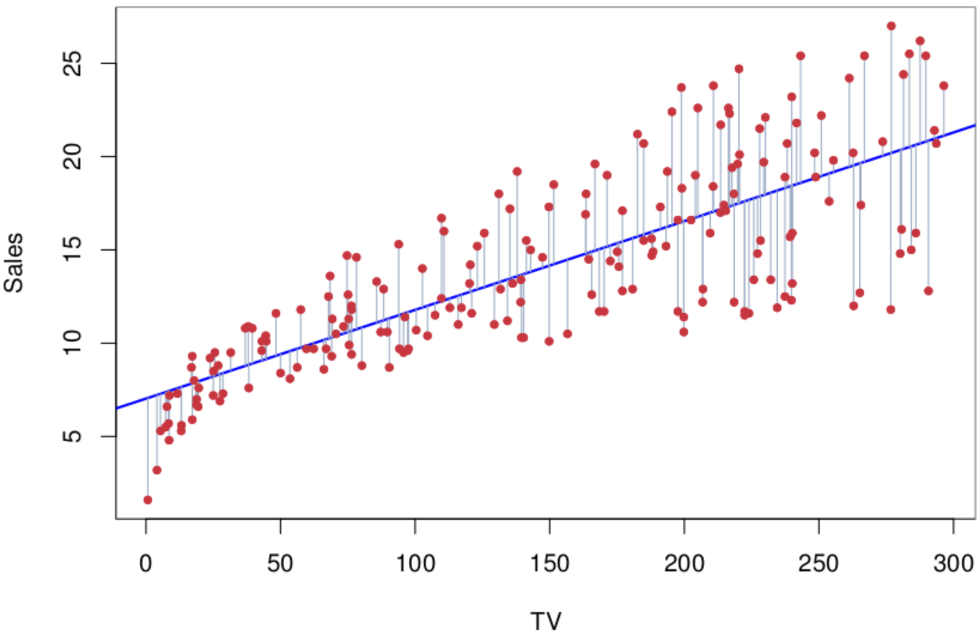


Figure 1: Linear Model for TV vs Sales

$e = y_i - \hat{y}_i$

Despite the aim of minimizing the absolute distances between the predicted and the actual values in practice we instead minimizes the *Residual Sum of Squares (RSS)*, as it is more mathematical well-behaved than absolute values.

The *residual sum of squares (RSS)* is calculated as:

$RSS = e_1^2 + e_2^2 + ... + e_n^2 <=>$
$RSS = (y_1 + \beta_0 - \beta_1 x_1)^2 + (y_2 + \beta_0 - \beta_1 x_2)^2 .... + (y_n + \beta_0 - \beta_1 x_n)^2$

The two *coefficients*, β_0 and β_1, can be minimized with the following equations:

$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})}$
$\beta_0 = \overline{y} - \beta_1 \overline{x}$

In **Figure 2** you can see the calculated RSS for different β_0 and β_1 values used on the advertisement dataset. The red dot marks the minimum RSS. The minimum can be found by differentiating the RSS model, and finding finding where it is equal to zero.

1.2 Assessing the Accuracy of the Coefficient Estimates

We have a true model:

$Y = 2 + 3X + \epsilon$

We create 100 random Xs and calculate Ys where we add some random noise, ϵ. In **Figure 3** these noisy points are shown, the true *population regression line* is shown in red and the *least squares line* calculated on the noisy sample is shown in blue.

To the right in **Figure 3** we have made noisy samples ten times and calculated their regression lines (light blue).

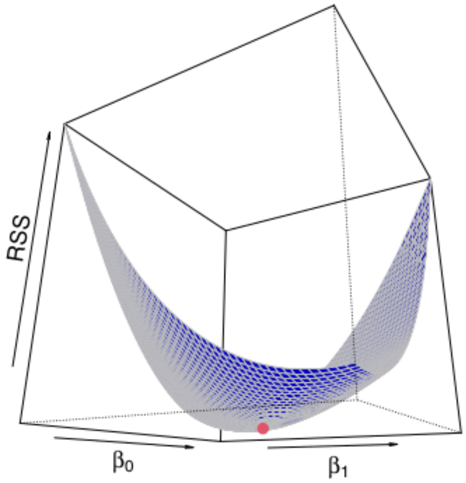


Figure 2: Minimizing RSS

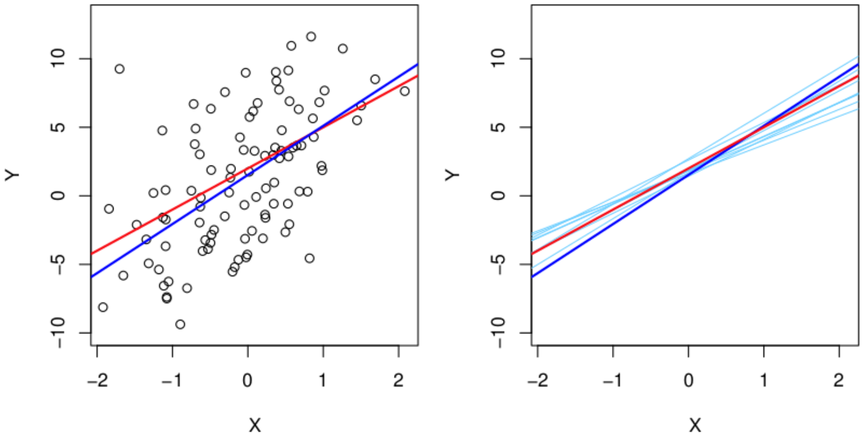


Figure 3: Regression Lines for Similated Data

In general the *sample coefficients* are good estimates of the *population coefficients*, and on average (μ) we expect them to be equal to the *population coefficients*. However, in a real world scenario we only have one sample, and we do not know if it is an over- or under-estimate of the *population coefficients*.

In general our confidence in the *sample coefficients* are estimated by computing the *standard error*. The *standard error* can be calculated via a formula (see below) or via *bootstrapping*. I prefer bootstrapping as it is a more transparent solution, as you sample your sample, and identify the deviation in the sample mean (standard error).

$Var(\hat{\mu}) = SE(\hat{\mu})^2 = \frac{\sigma^2}{n}$

The *standard error* roughly tells us the average amount that the estimated sample $\hat{\mu}$ differs from the actual value of μ. It is calculated in the following way:

1. Collect a sample of data from the population you're interested in studying.
2. Calculate the mean of your sample
3. Calculate the difference between each observation in your sample and the sample mean $(x - \overline{x})$
4. Square each of the differences obtained in Step 3.
5. Sum up all the squared deviations obtained in Step 4.
6. Divide the sum of squared deviations from Step 5 by the sample size (n), which gives us the variance (the average squared deviation).
7. Calculate the standard error (SE) by taking the square root of the variance (s^2) and dividing it by the square root of the sample size (n).

The more observations we have the smaller the standard error of $\hat{\mu}$.

The Standard errors can be used to compute *confidence intervals*. A *95 % confidence interval* e.g. is a range of values such that with 95 % probability the range will contain the true value.

For linear regression the 95 % confidence interval for β_1 can be approximated using:

$\hat{\beta_1} \pm 2 * SE(\hat{\beta_1})$

t-statistics is the number of standard deviations a value is away from zero.

$t = \frac{\hat{\beta_1 - 0}}{SE(\hat{\beta_1})}$

1.3 Assessing the Accuracy of the Model

We want to quantify *the extent to which the model fits the data*. This is usually assessed using the two quantities:

- R^2 Statistics
- Residual standard error (RSE)

1.3.1 R^2

R^2 is a measure of the *proportion of variance explained*. It is calculated in three steps:

1. Run a line through the mean, and calculate the *total sum of squares (TSS)*: $TSS = \sum(y_i - \overline{y})^2$.
2. Calculate the *residual sum of squares (RSS)* for your model: $RSS = \sum(y_i - \hat{y}_i)^2$.
3. Now we can calculate the proportion of the *variance that is explained by the model*: $R^2 = \frac{TSS - RSS}{TSS}$.

- Note that sometimes the formula includes all values divided by n as TSS/n is the *variance*, but during division n disappears, which is why we don't include it in the calculation here.

If a model fits the data perfectly the R^2 is 1. It should be noted that as more parameters are added to a model the closer R^2 will get to 1.

If a model is only built from two values R^2 will always be equal to one. So R^2 is not perfect, and we need a *p-value* to assess our confidence in R^2, which is done using the *F-statistic*. F is calculated in the following way:

$F = \frac{Variation\ in\ Y\ explained\ by\ X_1}{Variation\ in\ Y\ not\ explained\ by\ X_1} <=> F = \frac{TSS - RSS/(p_{fit} - p_{mean})}{RSS(n - p_{fit})}$

Here p_{fit} is the number of parameters in the model (y-intercept + slope), p_{mean} is the number of parameters in the mean line (y-intercepts) and n is the number of samples.

To estimate the *p-value* we calculate F for a bunch of random data sets (with similar variance and range), and get a *F-distribution*. Now we can calculate how many percent of the values in the *F-distribution* are equal or higher than the *F-value* we calculated for our model. This percentage is the *p-value*.

1.3.2 Residual Squared Error

The RSE is the average amount that the response will deviate from the true regression line. The formula uses a *magic number*, 2, to account for the error in small data sets.

$RSE = \sqrt{\frac{1}{n-2} RSS}$

We can calculate the percentage error by comparing the error if we had just predicted the mean value of y for all data points to the RSE. The *RSE/MeanError* tells us the percentage error.

2. Multiple Linear Regression

I practice we have more than one predictor, thus here we extend the *simple regression* to multiple variables.

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$

In such a model we interpret β_j as the average effect on Y of a one unit increase of X_j if all other predictors are fixed.

2.1 Estimating the Regression Coefficients

As with the simple linear regression we again want to minimize the RSS. See **Figure 4**.

The coefficients for a Multiple Linear Regression can be interpreted in the following way. For a given amount of X_1 and X_2 the spending an additional unit of X_3 is associated with β_3 units of Y.

If one had made three simple regressions and a single multiple regression model, the coefficients of the different X_j wouldn't necessarily be the same. This is because in the *multiple regression setting* the coefficient for one predictor represents the average increase in sales if the other predictors are held constant. This means that the effects of correlated predictors will be evened out, e.g. in a *simple regression* ice-cream sales would be associated with shark attacks, while in *multiple regression* where we add a predictor stating how many people are in the water, the effect of ice-cream sales on shark attacks will likely dissapear.

2.2 Is at least one of the predictors $X_1, X_2, ..., X_p$ useful in predicting the response?

We have to test if all the regression coefficients are zero.

$H_0 : \beta_1 = \beta_2 = ... = \beta_p = 0$

Versus the alternative

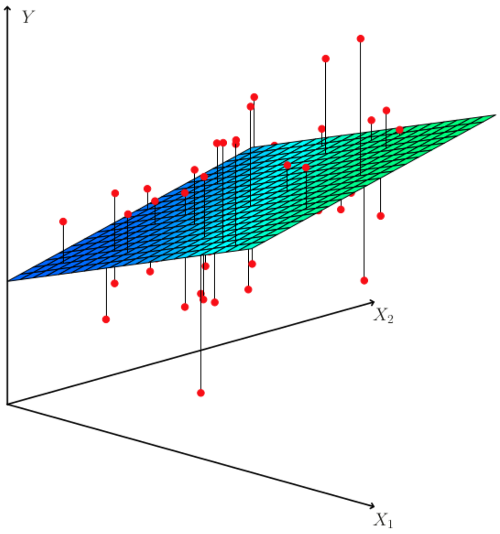


Figure 4: Multiple Linear Regression

$H_a : at\ least\ one\ \beta_j\ is\ non-zero$

This hypothesis test is performed by computing the *F-statistics*.

$F = \frac{(TSS-RSS)/p}{RSS/(n-p-1)}$

If there is no relationship, you would expect F to take on a value close to 1. If there is a relationship you would expect $E\{(TSS - RSS)/p\} > \sigma^2$. How large the F-statistic needs to be depends on the values of n and p. The significance (p-value) of the F-statistic is usually looked up for different n and p in a look-up table, which has values for the different F-distributions.

If we want to test the hypothesis that a subset of the coefficients are zero, we just fit another model that only uses that subset of variables.

The t-statistic and p-values for each model reports the *partial effect* of adding that variable to the model. That is, models are created with and without the variables, and their F-statistics are calculated in the following way:

$F = \frac{(RSS_o-RSS)/q}{RSS/(n-p-1)}$

Here RSS_0 is the value for the model without the variable, and q is the number of variables (1).

2.3 Do all the predictors help explain Y?

When computing the F-statistic p-value we determine if *at least one variable* is associated with the outcome, but now it is time to find which ones.

One could simply look at their p-values, but if p is large, one is likely to make false discoveries. What one instead should do is *variable selection*. One could try out all possible models and then judge their quality using one of the following: *Mallow's* C_p , *Akaike information criterion (AIC)*, *Bayesian information criterion (BIC)*, and *Adjusted* R^2.

It is, however, usually practically impossible to make all 2^p models, so instead usually does one of the following things:

- Forward Selection

1. Start with a *null model*
2. Fit one model with each predictor, and add the predictor with the lowest RSS
3. Repeat 2. until some stopping rule

- Backward Selection
    1. Start with all variables
    2. Remove the variable with the largest p-value
    3. Fit a new model
    4. Repeat 2. and 3. until a stopping rule is fulfilled
- Mixed Selection
    1. Start with a *null model*
    2. Fit one model with each predictor, and add the predictor with the lowest RSS
    3. Remove a variable if its p-value has gone over the level of significance.
    4. Repeat 2. and 3. until some stopping rule

2.4 How well does the model fit the data?

As with a *simple linear model* we again use RSE and R^2. One should, however, be aware that as the number of predictors increases the R^2 always increases! That is why one should look at the adjusted R^2 as it takes the extra parameters into account.

2.5 Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

Inaccuracies in the coefficient estimates can be computed using a *confidence interval* Inaccuracies in the Y estimates can be computed using a *prediction interval*.

3. Interactions

Linear models make several highly restrictive assumptions that are often violated in practice. Two of the most important assumptions state that the relationship between the predictors and response are additive and linear. Here we remove the *additive* assumption by adding an *interaction* term to the model.

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon <=> Y = \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \epsilon$

This can be interpreted as if adding one unit of X_1 then Y will increase by $\beta_1 + \beta_3 X_2$ units.

NOTE: The hierarchical principle states that if we include an interaction in a model, we hierarchical should also include the main effects, even if the p-values associated with principle their coefficients are not significant.

4. Non-Linear Relationships

Here we present a very simple way to directly extend the linear model to accommodate non-linear relationships, using *polynomial regression.* NOTE: It is still a *linear model*!

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon$

5. Common Problems with Linear Models

Here we go over the most common problems when fitting linear models to data:

5.1 Non-Linearity of the Data

If plotting the residuals against the fitted/predicted values it is possible pick up non-linearities as show in the figure. You would expect the errors to be equally distributed around zero for all Y-values if the model fits the data well.

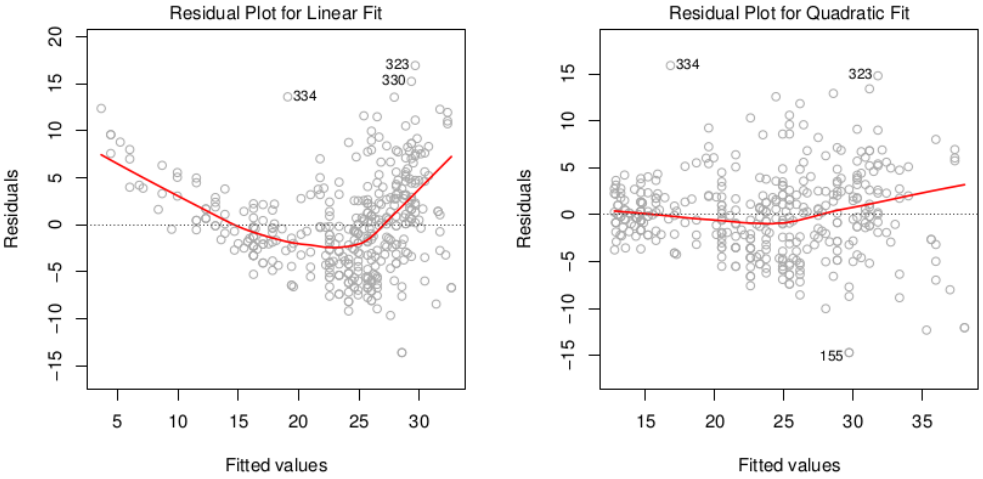


Figure 5: Residual Plot

Ways to adress the issue is to make non-linear transformations to the data (log X, $\sqrt{X}$, X^2).

5.2 Correlation of Error Terms

An important assumption of the linear regression model is that the error terms, ϵ_1, ϵ_2 , . . . , ϵ_n, are uncorrelated. If they are correlated:

- The estimated *standard errors* of the *regression coefficients* will be underestimated.
- Confidence and prediction intervals will be too narrow
- p-values will be too low

The problem is best understood with the example where we accidentally double our data, which would result in a narrowing our our confidence interval by a factor of $\sqrt{2}$. The problem is most often seen in time series where observations that are obtained at adjacent time points will have positively correlated errors. The problem can be identified by plotting the residuals as a function of time, and be seen if adjacent data points have similar residuals (this is called *tracking*).

This problem can mainly be mitigated by making a good *experimental design*.

5.3 Non-constant Variance of Error Terms

Another important assumption of the linear regression model is that the error terms have a constant variance, $Var(\epsilon_i) = \sigma^2$. The standard errors, confidence intervals, and hypothesis tests associated with the linear model rely upon this assumption.

The most common case of this is seen where the variance of the error terms increase as the value increases, this is also called *heteroscedasticity*. A example can be seen in the figure.

One of the most usual ways to fix this problem is to $logY$ or $\sqrt{Y}$. Another way is to use *weighted least squares*.

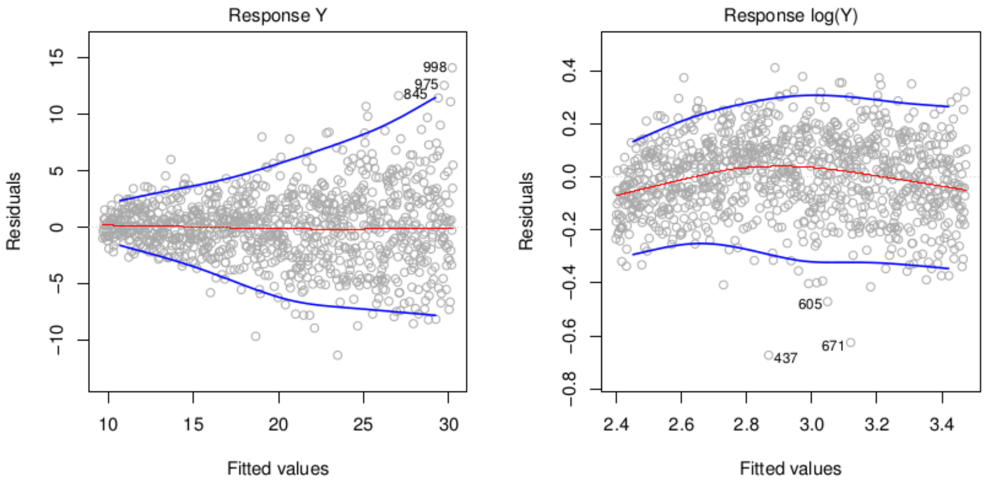


Figure 6: Heteroscedasticity

5.4 Outliers

An *outlier* is a point for which y_i is far from the y predicted by the model. Outliers can cause several problems.

1. It can result in a high RSE

- As RSE is used to calculate *confidence intervals* and *p-values* a single outlier can have implications for the interpretation of the fit.

2. It can result in a bad fit

- If it is a high *leverage point* the fit will be skewed towards that point

Outliers can be identified visually by plotting the residuals

5.5 High Leverage Points

These are points with unusual X-values, which means that they can have a sizable impact on the estimated regression line (see figure point 40). *High Leverage Points* are easy to visually identify in *simple linear regressions*, but much harder in *multiple regression* with many predictors, as it is possible to have an observation that is well within the range of each individual predictor's values, but is unusual in terms of the full set of predictors. In the figure you can see an example where neither the value for X1 nor its value for X2 is unusual, but where the combination of values is unusual.

One can, however, identify *high leverage* points using *leverage statistics*, where h_i increases withthe distance of x_i to $\overline{x}$.

$h_i = 1/n + \frac{(x_i - \overline{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \overline{x})^2}$

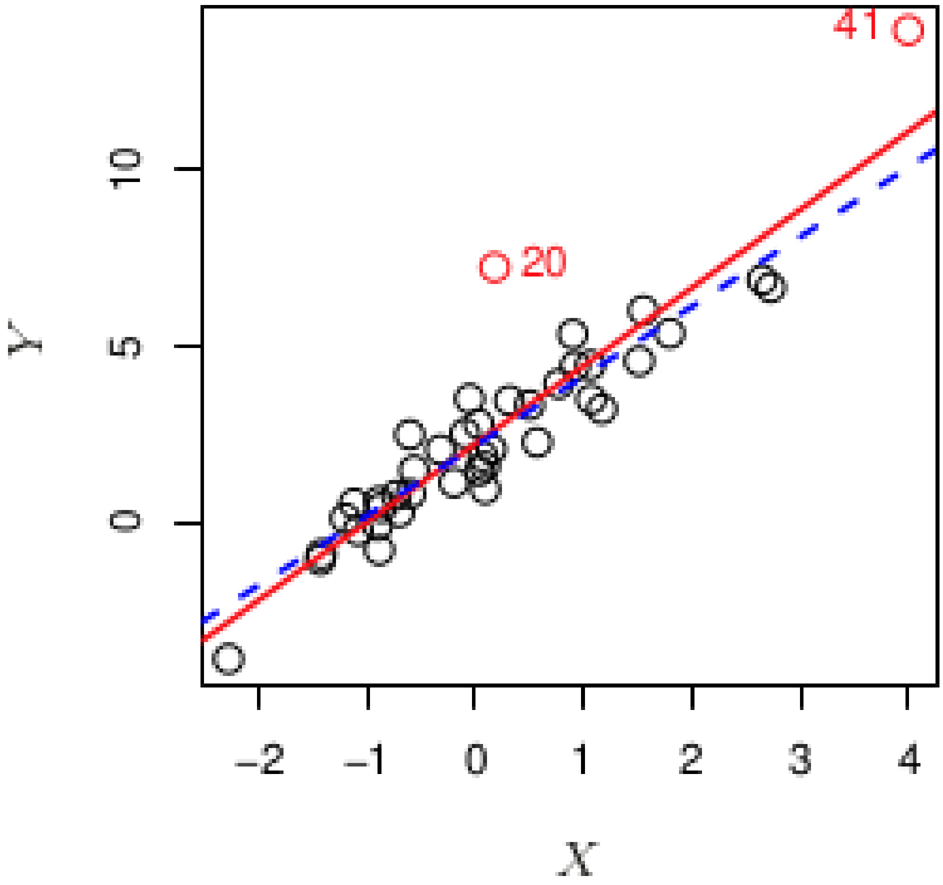


Figure 7: Leverage Points

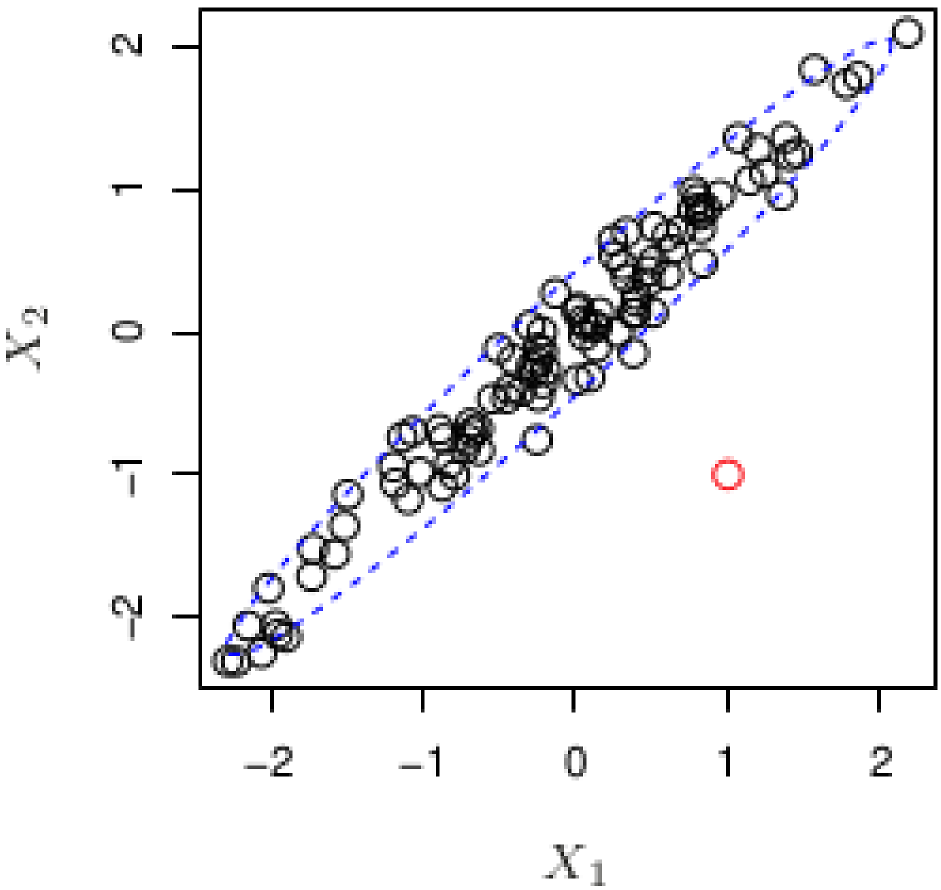


Figure 8: Leverage Points 2

5.6 Collinearity

Collinearity refers to the situation in which two or more predictor variables are closely correlated with each other. In a regression context it can be difficult to separate out the individual effects of the collinear variables on the response. The coefficients will have a high uncertaincy and thus so will the T-statistics.

It is even possible for collinearity to exist between three or more variables even if no pair of variables has a particularly high correlation. We call this situation *multicollinearity.* Instead of inspecting the correlation matrix, a better way to assess *multicollinearity* is to compute the variance inflation factor (VIF). A VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity.

Collinearity can be fixed by:

- Dropping one of the values
- Combining variables into a single predictor

6. Example of answering questions

- Is there a relationship between sales and advertising budget?
  - Fit *multiple regression*
  - Check F-statistics p-value
- How strong is the relationship?
  - Look at *RSE* and R^2
- Which media are associated with sales?
  - Examine the p-values associated with each predictor's t-statistic
- How large is the association between each medium and sales?
  - Compute *standard errors* for coefficients
  - Compute the 95 % confidence intervals for the coefficients
- How accurately can we predict future sales?
  - Calculate a prediction interval
- Is the relationship linear?
  - Make a *residual plot*
- Is there synergy among the advertising media?
  - Make an additive term at investigate its p-value
  - Does the R^2 get bigger?

7. Linear Regression vs KNN

Linear Regression Pros: * Easy to fit because only a small number of coefficients are estimated * Simple interpretations of coefficients * Tests of statistical significance can be easily performed Linear Regression Cons: * This may strong assumptions about the form of f(x)

KNN Pros: * Does not make specific assumptions about the form of f(x) * Handles non-linearities KNN Cons: * Can not extrapolate outside training data * Hard to chose the best K * Even when true relationships are non-linear it performs worse than Linear Regression models e.g. in *high dimensions*

8. Chapter Exercises & Answers

As most exercises are redundant for my brush up on statistical methods, I will only be solving select exercises. Great answers by Liam Morgan to all of the exercises can be found at RPubs.

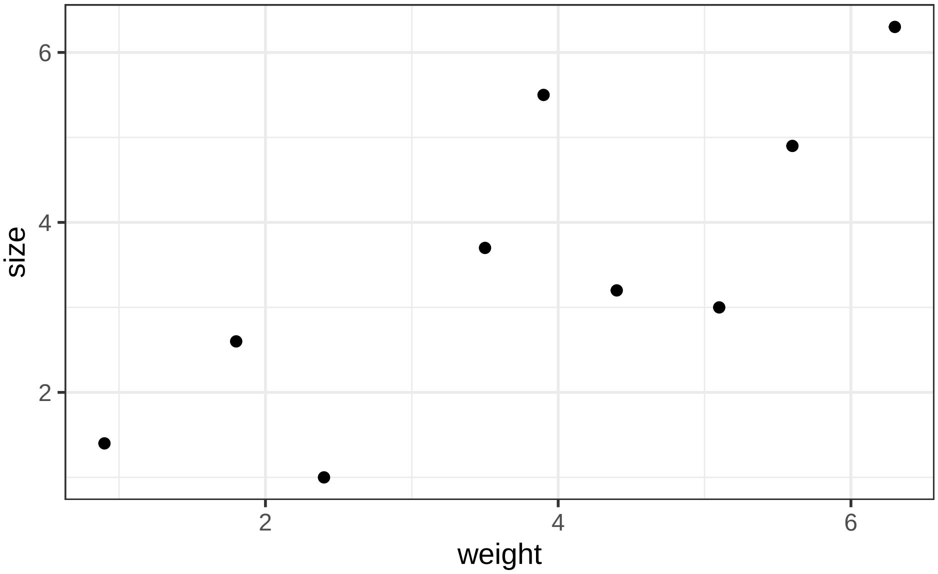
8.1 Simple Linear Models

Here we will just build a simple linear model and examine its different statistics. First we create some data.

```
# Libraries
library(tidyverse)

# Creating some data
mouse.data <- data.frame(
  weight = c(0.9, 1.8, 2.4, 3.5, 3.9, 4.4, 5.1, 5.6, 6.3),
  size = c(1.4, 2.6, 1.0, 3.7, 5.5, 3.2, 3.0, 4.9, 6.3)
)

# Plotting data
ggplot(mouse.data, aes(weight, size)) +
  geom_point() +
  theme_bw()
```



We create a simple linear model.

```
# Fitting a linear model
mouse.regression <- lm(size ~ weight, data=mouse.data)

# Summarizing model
summary(mouse.regression)
```

```
##
```

```
## Call:
## lm(formula = size ~ weight, data = mouse.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5482 -0.8037  0.1186  0.6186  1.8852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5813     0.9647   0.603   0.5658
## weight        0.7778     0.2334   3.332   0.0126 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.19 on 7 degrees of freedom
## Multiple R-squared:  0.6133, Adjusted R-squared:  0.558
## F-statistic:  11.1 on 1 and 7 DF,  p-value: 0.01256
```

Here we go over the summary of the model:

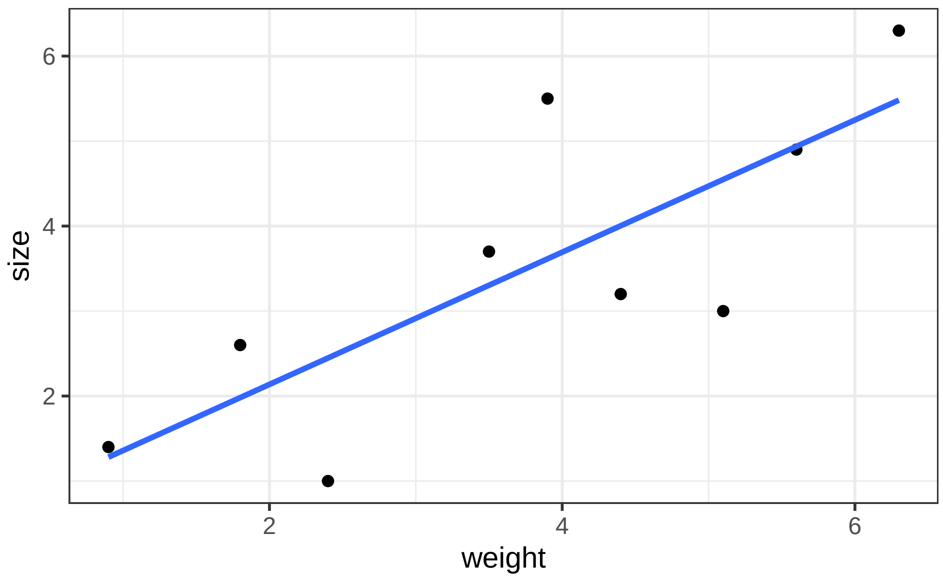
- Call: It is just the model call
- Residuals: The distribution of the residuals
  - You would like the distribution to be symmetric around zero
- Coefficients:
  - Intercept: The estimated intercept with the y-axis
    - Standard Error and t-values are used to calculate the p-value
    - p-value is an estimate of whether the value is zero or not
  - Weight: The estimated slope of the linear line
    - Standard Error and t-values are used to calculate the p-value
    - p-value is an estimate of whether the value is zero or not
- Residual Standard Error: Square root of the denominator for F
- Multiple R-squared: R-squared as described earlier
- Adjusted R-squared: R-squared scaled by the number of parameters
- F-statistic: Tells is if R-squared is significant or not

```
# Making predictions with confidence intervals
predict(mouse.regression,
        data.frame(weight = c(2,4,6)),
        interval = "prediction")
```

```
##        fit        lwr      upr
## 1 2.136962 -0.9862262 5.260149
## 2 3.692603  0.7227867 6.662418
## 3 5.248244  2.0352816 8.461206
```

Finally we add the regression line to our plot

```
# Plotting regression line
ggplot(mouse.data, aes(weight, size)) +
  geom_point() +
  geom_smooth(method = "lm",se = F) +
  theme_bw()
```

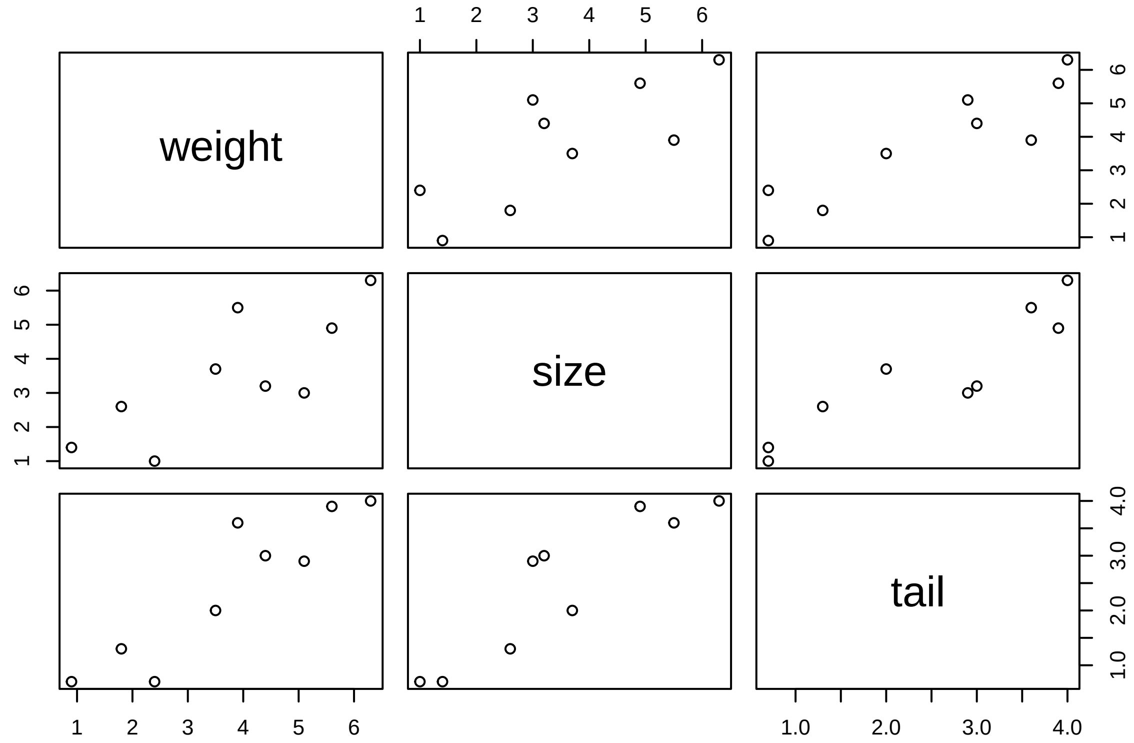


8.2 Multiple Regression

Here we build a multiple regression model and examine its different statistics. First we create some data.

```
# Creating some data
mouse.data <- data.frame(
  weight = c(0.9, 1.8, 2.4, 3.5, 3.9, 4.4, 5.1, 5.6, 6.3),
  size = c(1.4, 2.6, 1.0, 3.7, 5.5, 3.2, 3.0, 4.9, 6.3),
  tail = c(0.7, 1.3, 0.7, 2.0, 3.6, 3.0, 2.9, 3.9, 4.0)
)

# Plotting the data
plot(mouse.data)
```



In the plots we can see that both weight and tail are correlated with size, which is what we want, but we also see that they are correlated with each other. This may mean that a model with either tail or weight will be sufficient.

We create a multiple regression model.

```
# Fitting a plane to the data
mouse.regression <- lm(size ~ weight + tail, data=mouse.data)

# Summarizing model
summary(mouse.regression)
```

```
##
## Call:
## lm(formula = size ~ weight + tail, data = mouse.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.99928 -0.38648 -0.06967  0.34454  1.07932
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.7070     0.6510   1.086   0.3192
## weight       -0.3293     0.3933  -0.837   0.4345
## tail          1.6470     0.5363   3.071   0.0219 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.8017 on 6 degrees of freedom
## Multiple R-squared:  0.8496, Adjusted R-squared:  0.7995
## F-statistic: 16.95 on 2 and 6 DF,  p-value: 0.003399
```

As we are running multiple regression we are more interested in the *Adjusted* R^2, as it takes the extra parameter into account. We can see that the *p-value* for the weight is no longer significant.

When looking at the statistics for the coefficients they are based on a comparison between a model with the variable and without it. The p-value of 0.4345 means that a model that uses the weight and tail variables is not significantly better than one using just the tail variable.

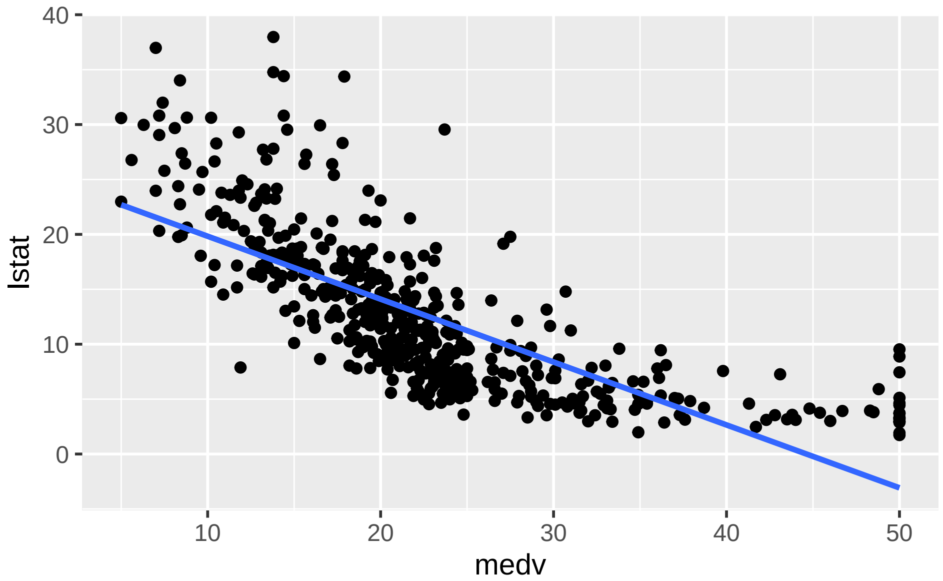
8.3 The Boston Dataset

Here we start simple and go deeper and deeper on the **Boston** dataset.

8.3.1 Fitting a simple model

```
# Loading Data
library(ISLR2)
library(car)

# Plotting data
ggplot(Boston, aes(medv, lstat)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

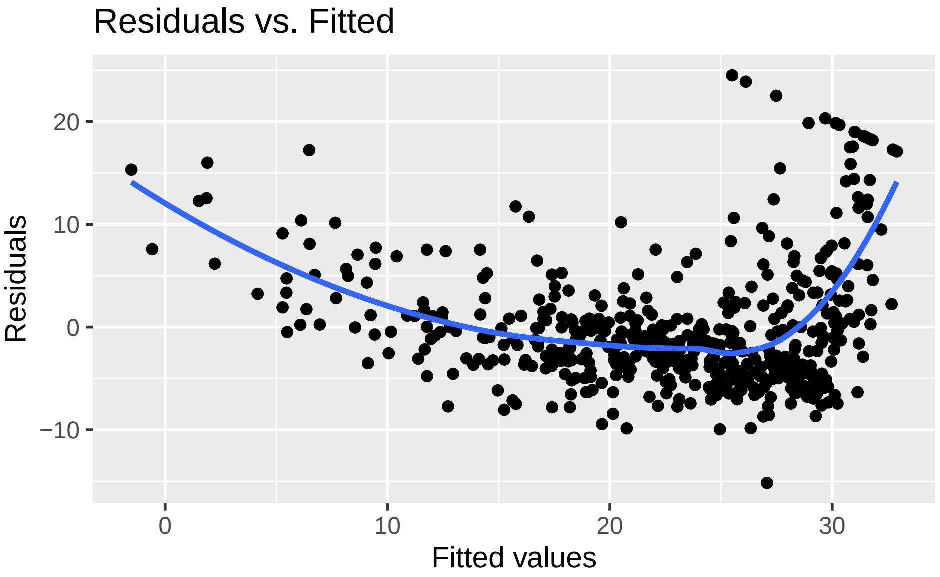


```
# Linear model
boston.lr <- lm(medv ~ lstat, data = Boston)
```

Now we will look at some diagnostics plots.

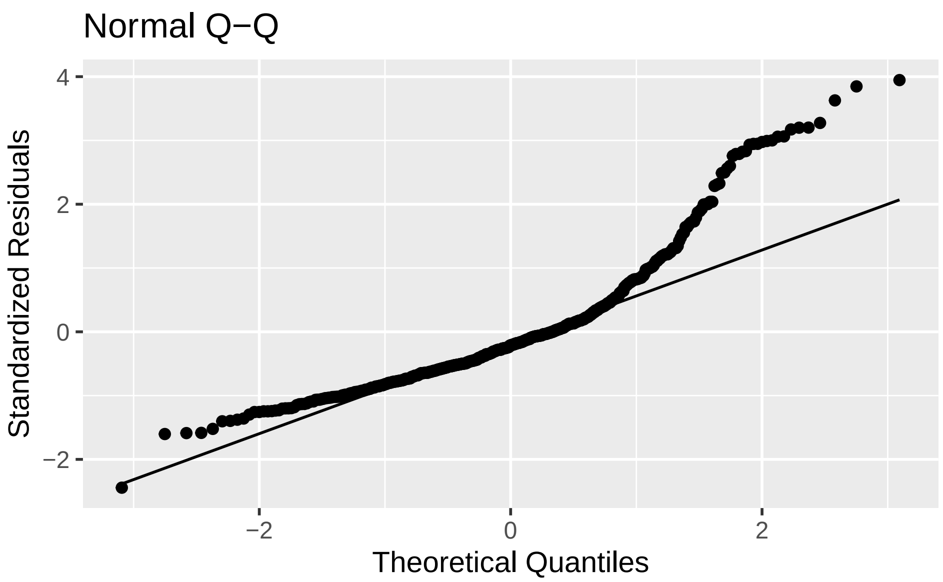
```
# Extract model information for plotting
data <- fortify(boston.lr)

# Residuals vs. Fitted plot
ggplot(data, aes(.fitted, .resid)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  xlab("Fitted values") +
  ylab("Residuals") +
  ggtitle("Residuals vs. Fitted")
```



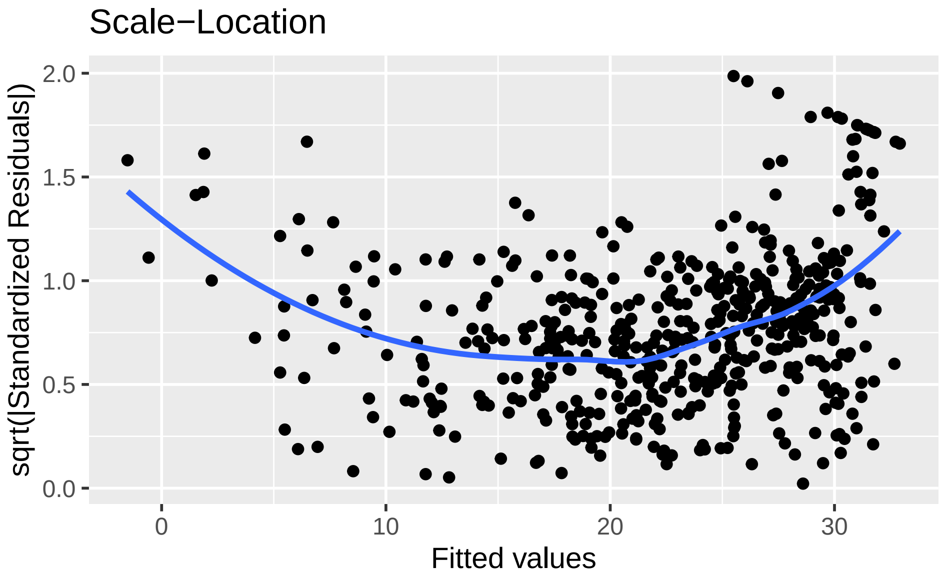
We can see that the data points are not evenly distributed above and below the line indicating that the data has a non-linear relationship.

```
# Normal Q-Q plot
ggplot(data, aes(sample = .stdresid)) +
  stat_qq() +
  stat_qq_line() +
  xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q")
```



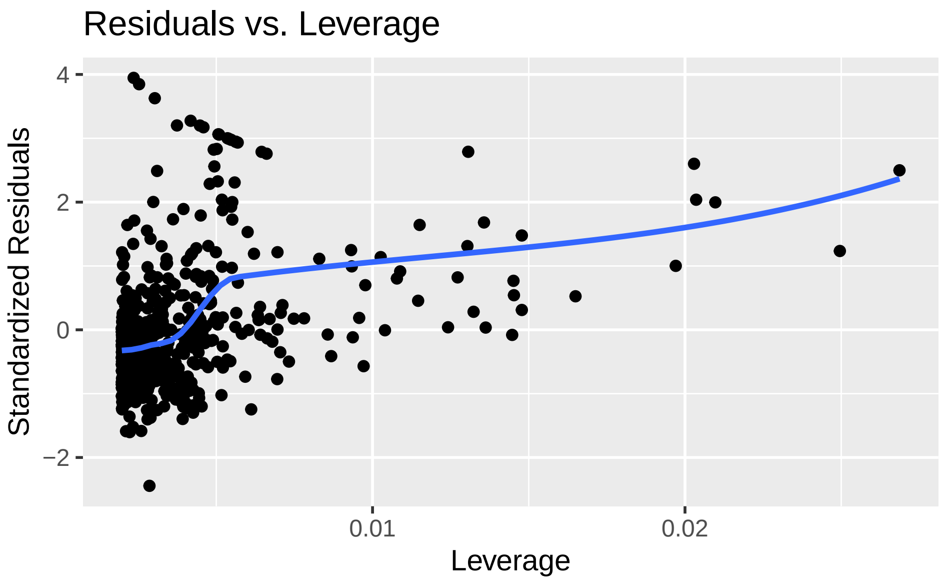
We can see that the residuals are not normally distributed though one would assume that.

```
# Scale-Location plot
ggplot(data, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  xlab("Fitted values") +
  ylab("sqrt(|Standardized Residuals|)") +
  ggtitle("Scale-Location")
```



Also known as the spread-location plot, it helps evaluate if the residuals have a constant variance (homoscedasticity). It plots the absolute values of the square root of the standardized residuals against the fitted values. We see a clear trend in the standardized residuals.

```r
# Residuals vs. Leverage plot
ggplot(data, aes(.hat, .stdresid)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +
  ggtitle("Residuals vs. Leverage")
```



This plot helps identify influential observations that have a large impact on the regression results. It displays the standardized residuals against the leverage, which measures how much an observation affects the model's fit.

8.3.2 Fitting a multiple regression model

Now it is time to include some more variables!

```r
# Fitting the model
boston.mr <- lm(medv ~ ., data = Boston)

# Summarizing statistics
summary(boston.mr)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.798 on 493 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

To assess the colliniarity next we calculate the *variance inflation factors*.

```
# Calculating VIFs
vif(boston.mr)
```

```
##     crim       zn    indus     chas      nox       rm      age      dis
## 1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037
##      rad      tax  ptratio    lstat
## 7.445301 9.002158 1.797060 2.870777
```

VIFs help quantify the extent of multicollinearity by measuring how much the variance of an estimated regression coefficient is inflated due to the presence of correlated predictors.

The VIF for a particular predictor variable is calculated as follows:

1. Fit a regression model using the predictor variable of interest as the response variable and the remaining predictor variables as the predictors.
2. Calculate the R-squared value of this regression model.
3. Calculate the VIF as the reciprocal of (1 - R-squared).

The interpretation of VIF values is as follows:

1. A VIF of 1 indicates no correlation between the predictor variable and the other predictors. There is no multicollinearity issue.
2. A VIF greater than 1 and less than 5 suggests moderate correlation among the predictors. It is generally considered acceptable.
3. A VIF greater than 5 indicates high correlation between the predictor variable and the other predictors, indicating significant multicollinearity.

8.3.3 Adding the interaction term

```r
# Adding an interaction term
summary(lm(medv ~ lstat * age, data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

8.3.4 Going Non-Linear

Here we make a model with and without a non-linearity.

```r
# Model
model.linear <- lm(medv ~ lstat, data=Boston)

# Model with non-linearity
model.nonlinear <- lm(medv ~ lstat + I(lstat^2), data=Boston)
```

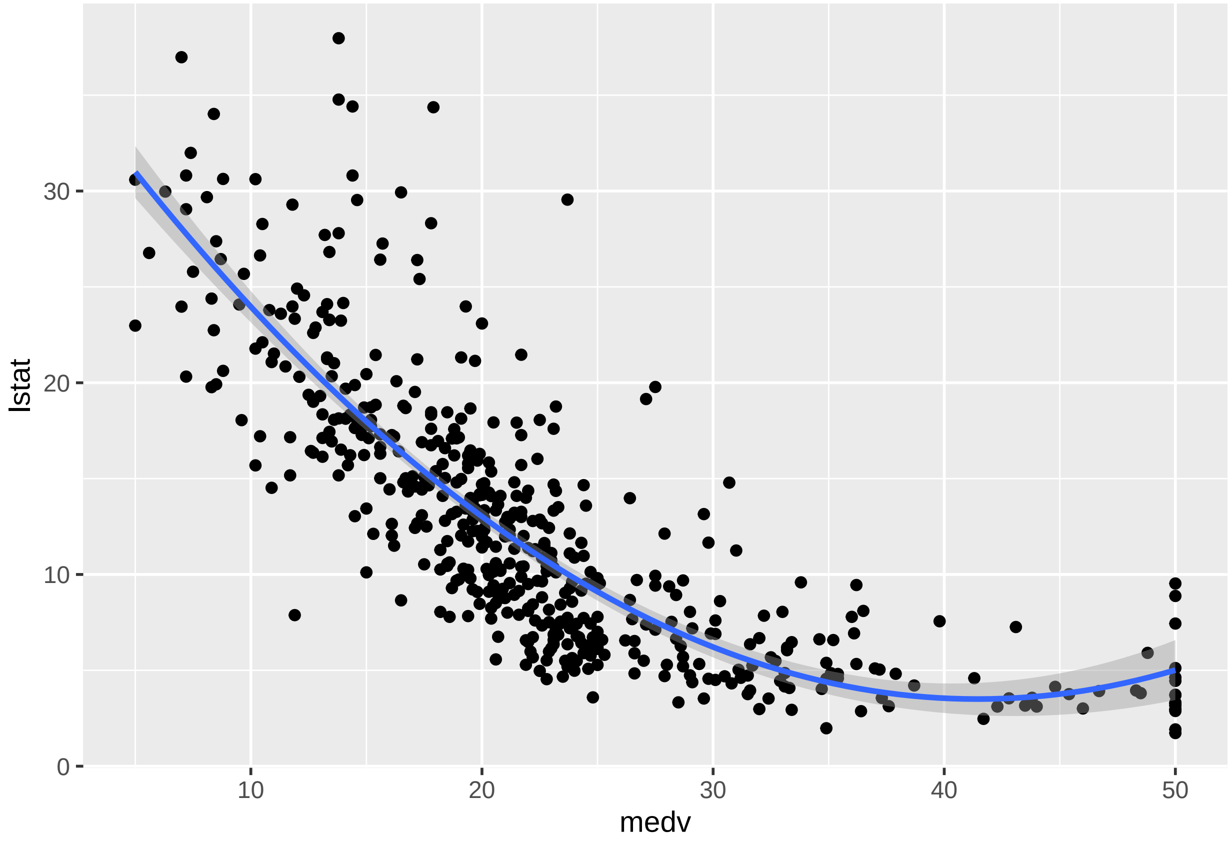
We can compare the two models using ANOVA. The null hypothesis is that the two models fit the data equally well, and the alternative hypothesis is that the full model is superior.

```r
# Testing if the model with non-linearity fits better
anova(model.linear, model.nonlinear)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic is 135, and the p-value is very low, thus the null hypothesis is rejected. Let try to plot the new line.

```
# Plotting data
ggplot(Boston, aes(medv, lstat)) +
  geom_point() +
  geom_smooth(method='lm', formula= y~x + I(x^2))
```


lstat
medv