Model assumptions

Introduction

When undertaken simple or multiple regression modeling, we make certain assumptions about the relationships between variables and assumptions about the residuals:

- There is a linear relationship between the exposure (or explanatory) variable and the outcome variable
- The residual values are:
  - Normally distributed
  - Homoscedastic
  - Independent

So before we carry on we do need to take a closer look at the idea of "residual" values.

> ❗ Important
>
> Our linear model provides us with a theoretical relationship between our variables that is represented by a straight line. As one variable changes, the other changes predictably along that line. The 'least squared' line is the closest fit to the actual data points. Note that most data in reality won't sit exactly on that line (the model). For each data point there will be some distance between it and the model. **We cal that distance the 'residual'.**

Let's take a look at the mtcars dataset, and in particular the relationship between the `weight` of cars and their fuel efficiency as measured by miles per gallon (`mpg`):

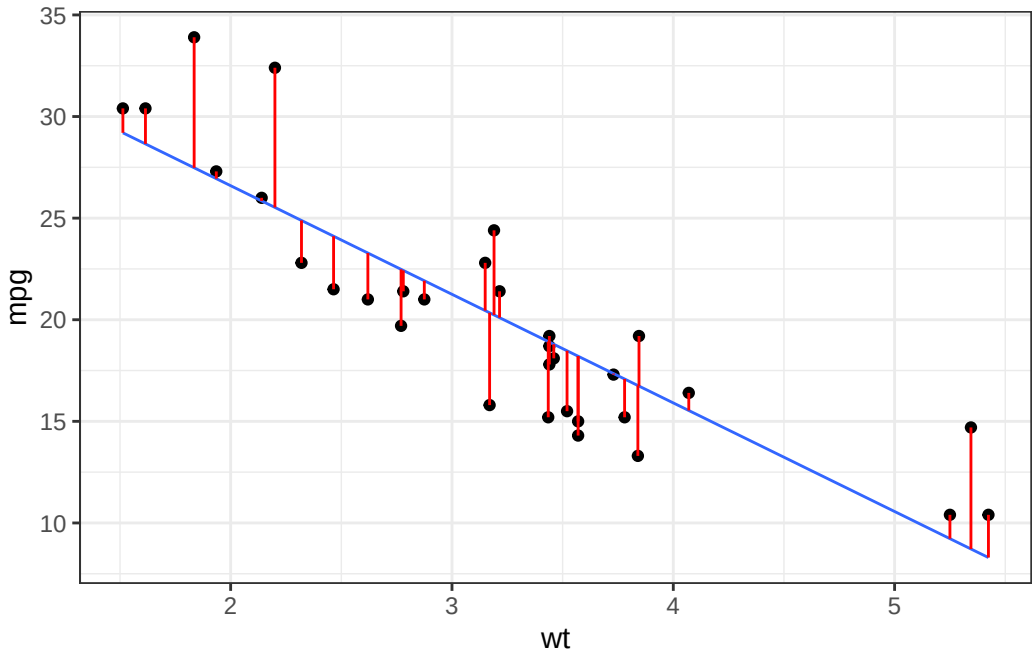
```
mtcars %>%
    select(mpg, wt) %>%
    head(5)
```

```
                   mpg    wt
Mazda RX4         21.0 2.620
Mazda RX4 Wag     21.0 2.875
Datsun 710        22.8 2.320
Hornet 4 Drive    21.4 3.215
Hornet Sportabout 18.7 3.440
```

Below, we'll plot the relationship between `wt` and `mpg` and then illustrate the residuals (the distance from each point to the model).

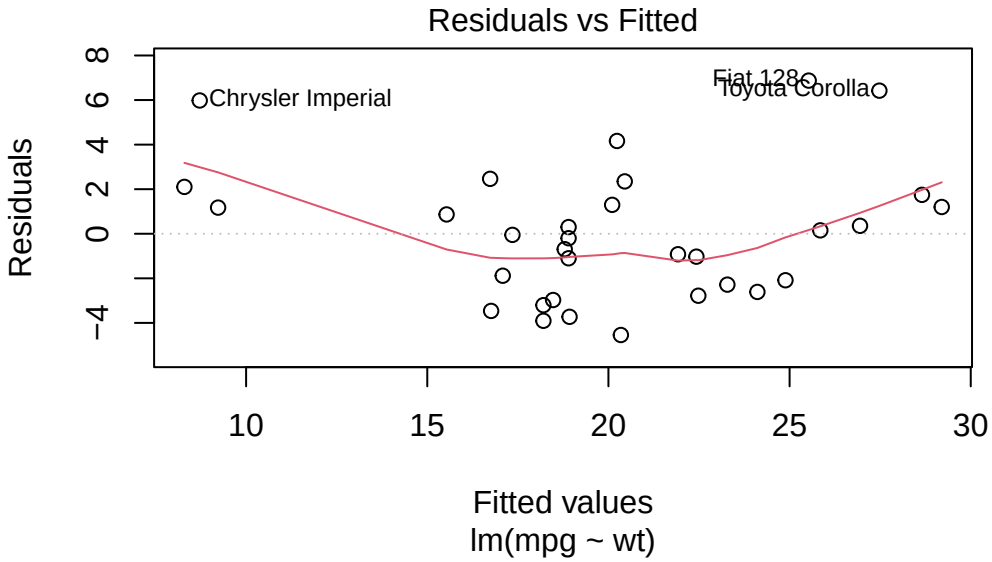
```
mtcars %>%
    mutate(predicted = predict(lm(mpg ~ wt, data = .))) %>%
    ggplot(aes(wt, mpg))+
    geom_point(size = 1.5)+
    geom_smooth(method = lm, se = F, size = 0.5) +
    geom_segment(aes(xend = wt, yend = predicted), color = "red")+
    theme_bw()
```

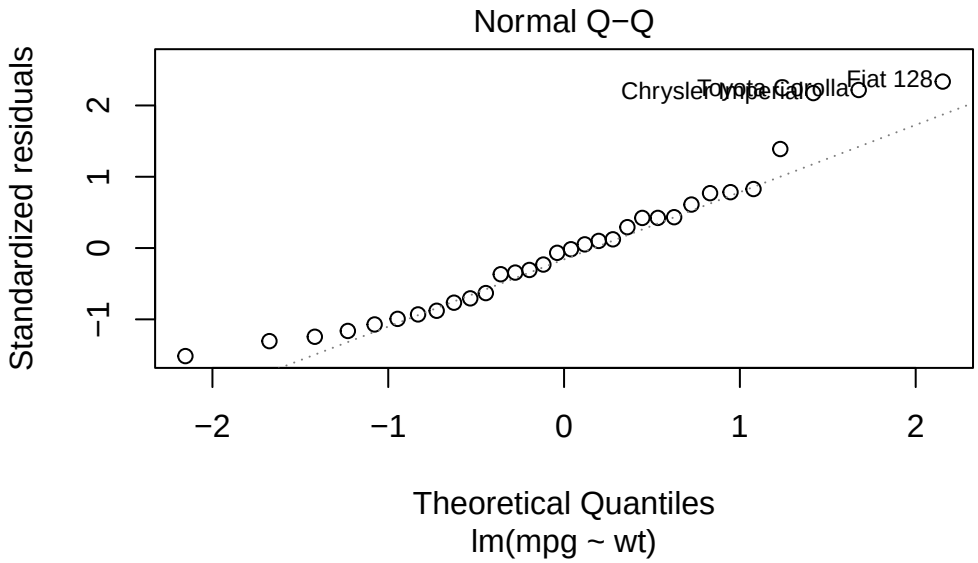
```
`geom_smooth()` using formula = 'y ~ x'
```

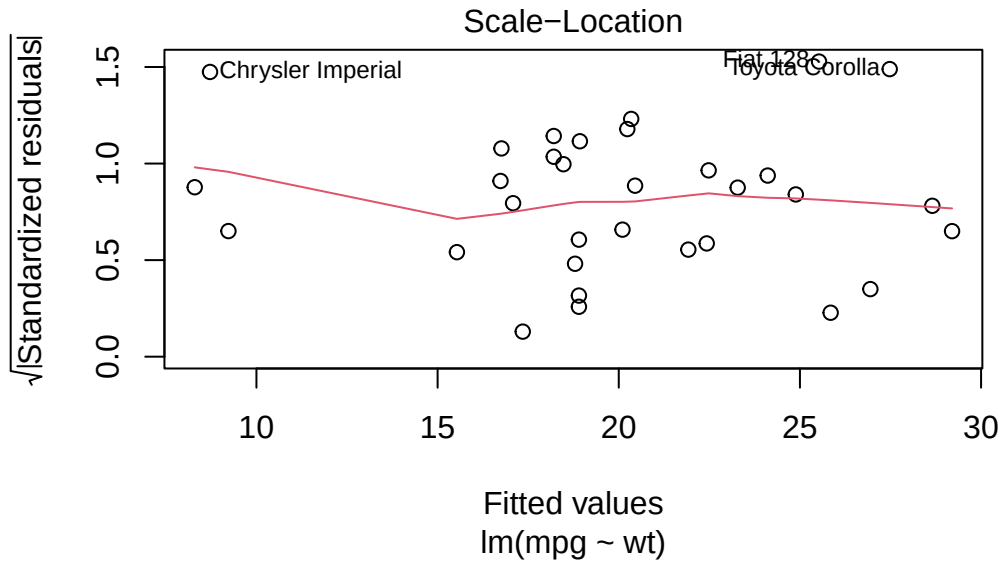


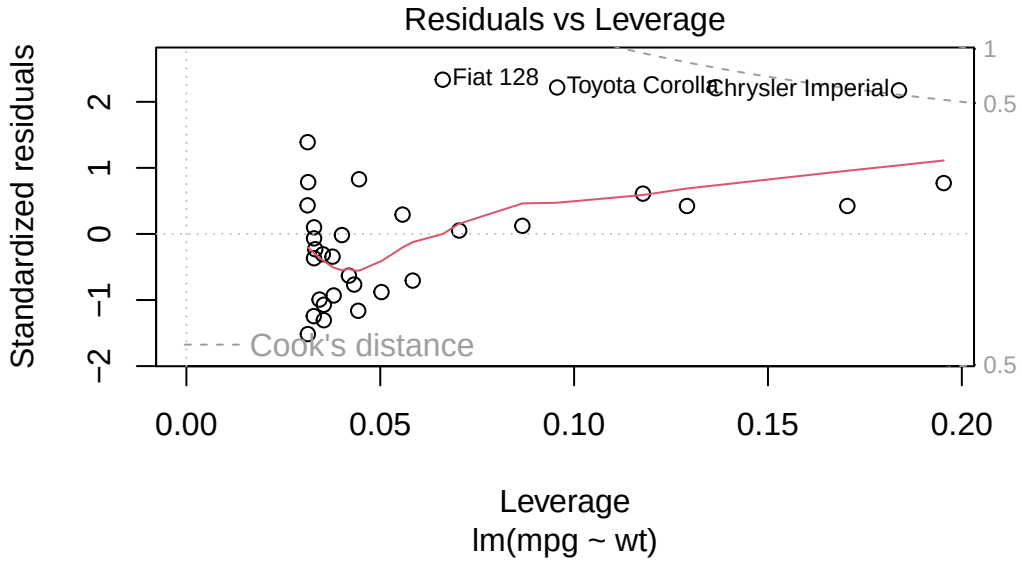
We can create diagnostic plots using `plot()` and include as the argument the model that we've created:

```r
model <- lm(mpg ~ wt, data = mtcars)
plot(model)
```









The red lines represent the residuals - the distance between the observed values and the values that the mode predicts.

So now that we understand what the residuals are, let's take a look at some of the assumptions that need to be true for a model to be useful.

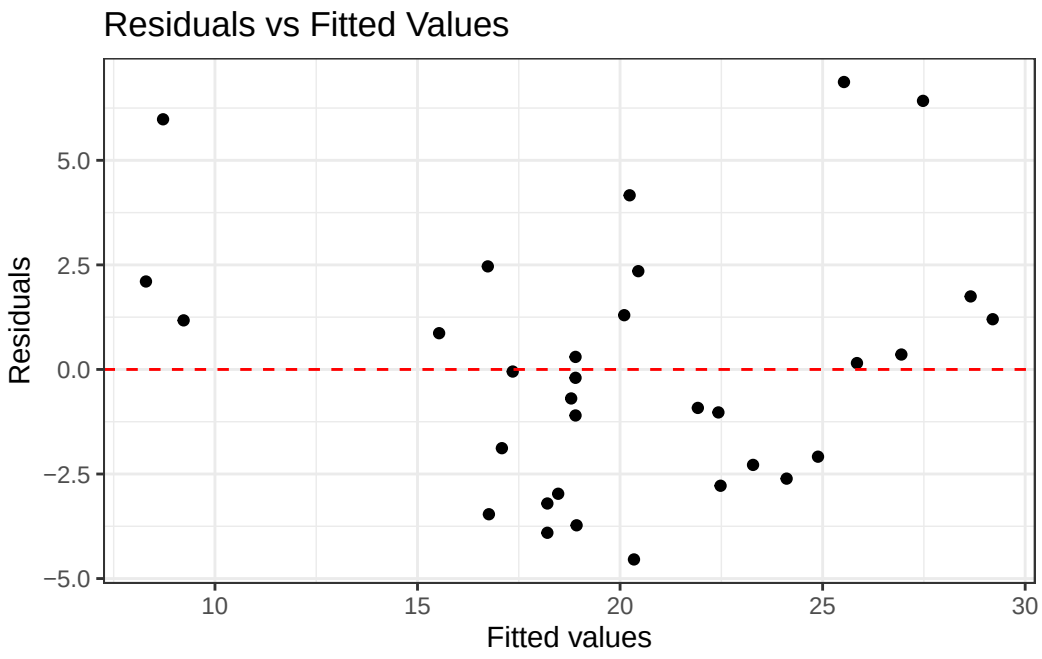
Assumptions

1. A linear relationship between the explanatory and outcome variable

In the case of the `mtcars` dataset, we need to be sure that there is a linear relationship between the `wt` and `mpg`.

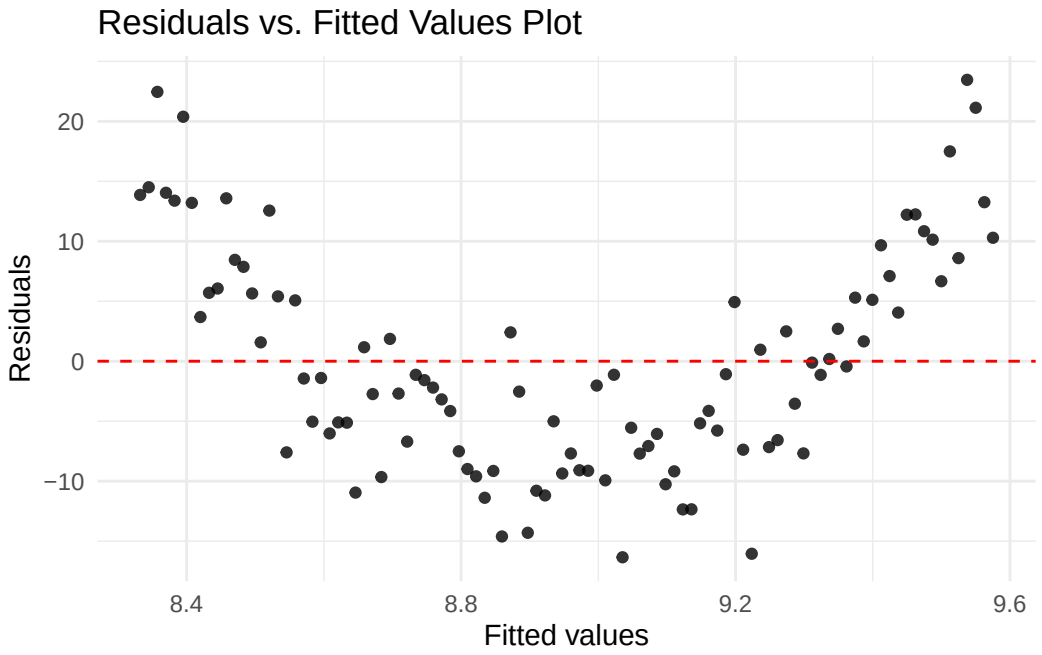
The first (and often only needed) method is to visualize the data, as we have above. Looking at the plot it might be clear that there is a linear relationship. A second visual check is to plot the fitted values (x axis) against the residuals (y axis).

```
mtcars %>%
    mutate(fitted = fitted(model)) %>%
    mutate(residuals = residuals(model)) %>%
    ggplot(aes(fitted, residuals))+
    geom_point()+
    geom_hline(yintercept = 0, color = "red", linetype = "dashed") +  # horizontal line at
  theme_bw() +
  labs(title = "Residuals vs Fitted Values",
       x = "Fitted values",
       y = "Residuals")
```



The residuals should be scattered randomly around the horizontal axis (which represents a residual value of 0). If the points are symmetrically distributed around a horizontal line without distinct patterns, that's a good sign of linearity. If you see a systematic pattern or a curve in the residuals, it suggests that the relationship between the predictor(s) and the response might be non-linear. For instance, a U-shaped or inverted U-shaped pattern often suggests missing polynomial terms (e.g., squared terms) in the model. The methods above can be subject to personal interpretation.

Here is an example of what a Residual vs Fitted value plot with a pattern would look like:



The correlation coefficient can also be looked at. Again there is not definitive cut off for what is considered to be sufficient but in general if the correlation coefficient is more than 0.3 (or less than -0.3 if negatively correlated) then there is moderate evidence of linearity.

```
cor(mtcars$wt, mtcars$mpg)
```

```
[1] -0.8676594
```

In this case the correlation coefficient strongly suggests linearity.

A statistically rigorous method of establishing linearity is to use the Harvey-Collier Multiplier Test for Linearity. To do this we'll need to install and call a the `lmtest` package:

```
#install.packages("lmtest")
library(lmtest)
```

We have already created the fitted model (above). All we need to do now is apply the `harvtest()` function.

```
harvtest(model)
```

```
    Harvey-Collier test

data:  model
HC = 0.23045, df = 29, p-value = 0.8194
```

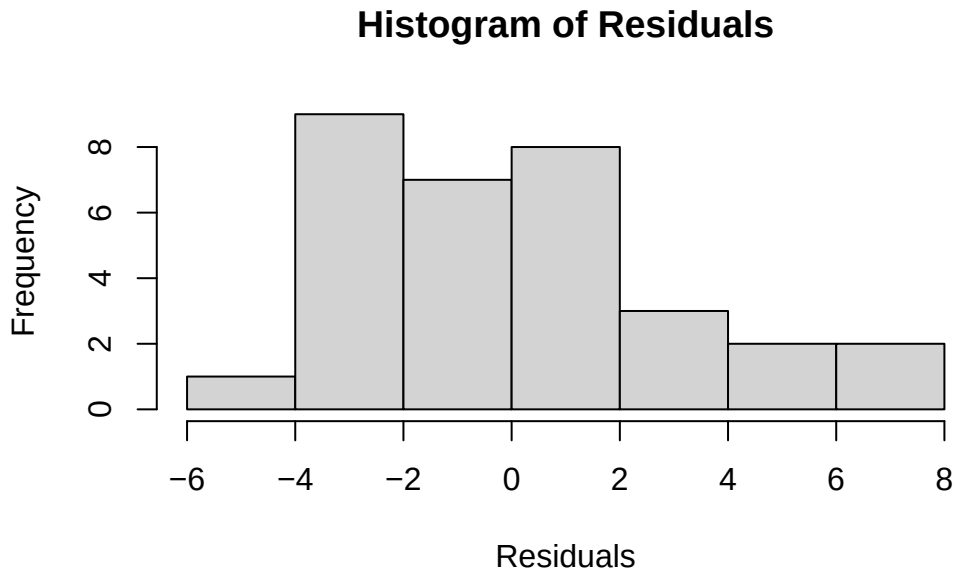
The interpretation of this test result is a little unusual. The null hypothesis is that there is a linear relationship. So a small p value (less than 0.05) would cause us to reject the assumption that the relationship between the two variables is linear. We're looking for a p value larger than 0.05 (as is the case in this model).

2. The residuals follow a normal distribution.

Remember, the residuals are the distance between the actual and predicted values. If there are any major outliers in the data or an unusual relationship between the predictor and outcome variables in certain intervals (perhaps related to a third variable) then the distribution of residual values won't be normally distributed.

Firstly we can visualize the residuals as a histogram:

```
hist(residuals(model),
     main = "Histogram of Residuals",
     xlab = "Residuals")
```



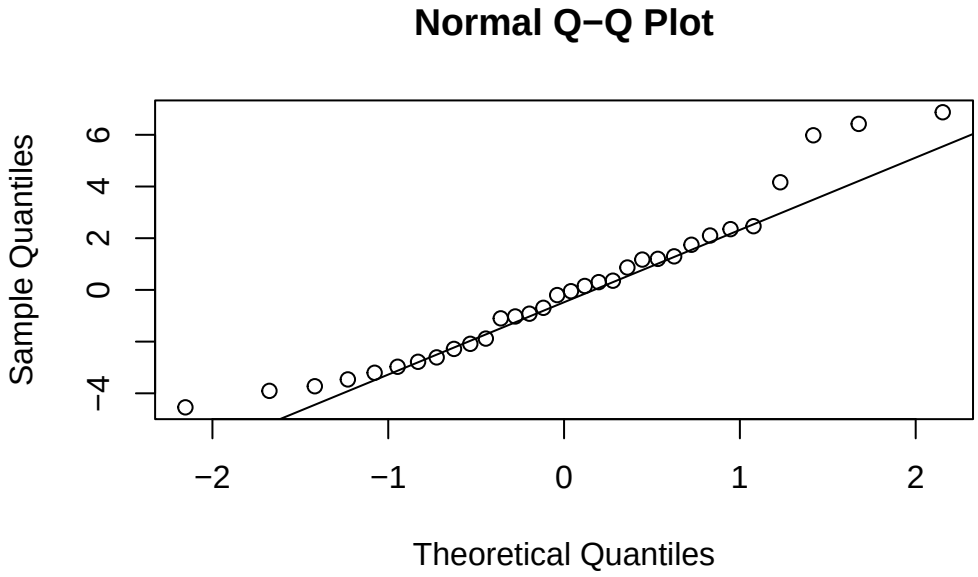
We can also use a Q-Q plot (Quantile - Quantile plot) is a visual tool used in statistics to help assess if a dataset follows a certain distribution, usually a normal distribution.

- **Quantiles**: These are points that divide the data into intervals with equal probabilities. For example, in a set of ordered data, the median splits the data into two halves where each half is 50% of the data. The median is a quantile.
- **The Plot**: In a QQ plot, you plot the quantiles of your data against the quantiles of a theoretical distribution, such as the normal distribution. Essentially, you're matching up values from your dataset against a perfectly normal distribution to see how well they line up.

We want to know if the residual values follow a normal distribution. The residual values can be extracted using the `residuals(model)`. Let's take a look.

```
qqnorm(residuals(model))
qqline(residuals(model), color = "red")
```

```
Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "color" is
not a graphical parameter
```



Again this requires subjective interpretation. We can use the Shapiro-Wilk test as a formal test of normality. The null hypothesis of this test is that the data is normally distributed. If the *p*-value is less than a chosen alpha level (commonly 0.05), the null hypothesis is rejected, indicating that the data deviates from a normal distribution.

```
shapiro.test(residuals(model))
```

```
    Shapiro-Wilk normality test

data:  residuals(model)
W = 0.94508, p-value = 0.1044
```

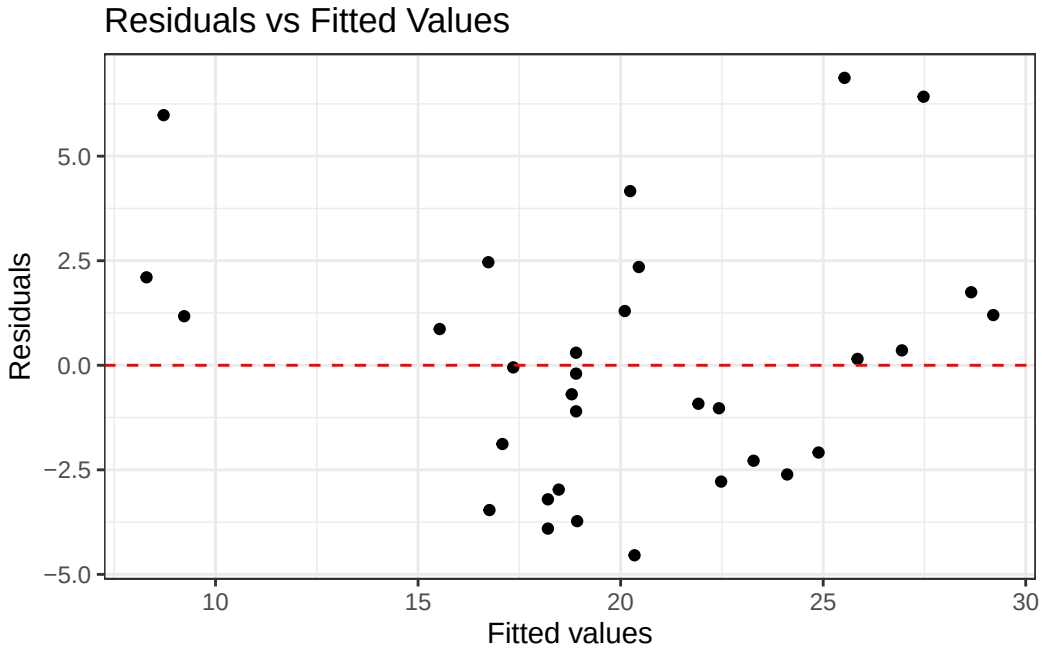
In this case the p value is large. We usually like small p values but not in this case. For the Shapiro-Wilk test a small p value suggests that the residuals are not normally distributed. A large p value (larger than the alpha value cut off of 0.05, for example) suggests that the residuals are normally distributed and our model assumption is met.

3. Residuals are homoscedastic

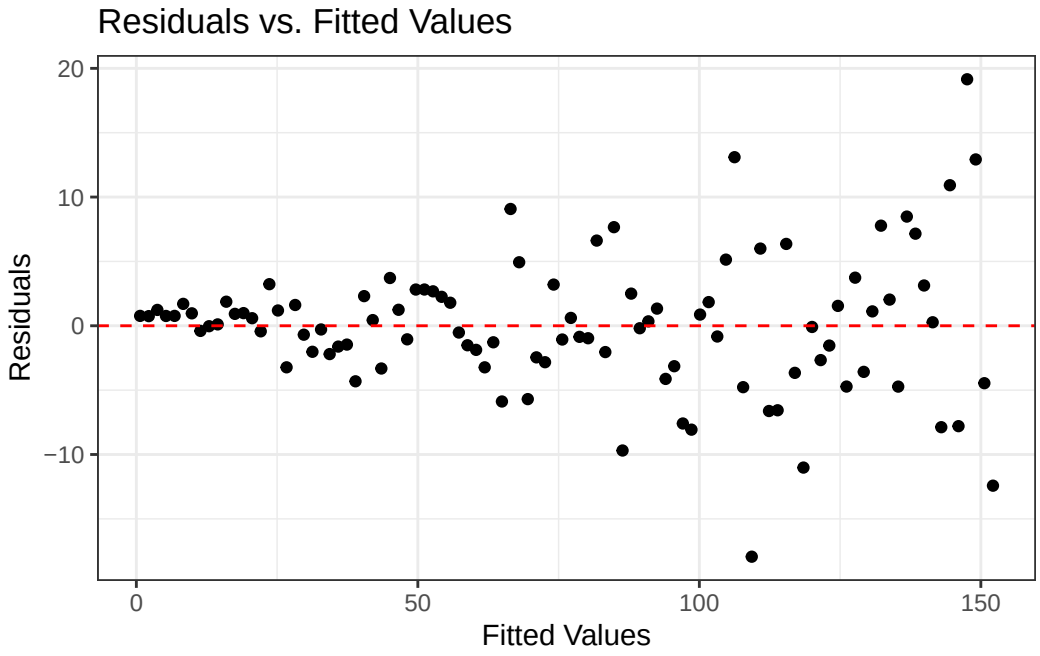
This refers to the assumption in regression analysis that the variance of the residuals is constant across all levels of the explanatory variables.

To illustrate this, think about the case of the mtcars dataset: we want the model residuals to vary from one point to next in a way that is consistent as we look at the graph from left to right (or across the values of the x-axis).

By plotting the fitted values against the residuals, one can look a pattern in the plot. This could be a funnel shape or clumping of the data points. This suggests that there is a problem and the residuals are not homoscedasctic (the assumption is not met). In chase of the mtcars dataset, we've already created this plot and as you can see below, the funnel shape doesn't appear.



So that you know what it is that you're looking for (and can recognize the funnel shape, here is an example of the assumption not being met:



There are other reasons for seeing unusual patterns in the plot of residuals vs fitted values (like non-linearity) so it is important to look at the whole picture when interpreting these plots.

While visual confirmation might be sufficient, you can use formal statistical methods to test for heteroscedasticity. The Breusch-Pagan test (or the Cook-Weisberg test) can be used.

We've already installed and called the `lmtest` package. The tests can be performed as follows:

```
bptest(model)
```

```
    studentized Breusch-Pagan test

data:  model
BP = 0.040438, df = 1, p-value = 0.8406
```

The null hypothesis assumes homoscedasticity. In other words, you'd like to see a p value of more than 0.05 to support the inclusion of the variable in your model. In this case we're very happy with a p value of 0.84Assumption: residual values are independent.

4. The residuals are independent

The key point is that for a given explanatory variable, each observation's residual should be independent of the residuals of other observations for that variable. This is easiest to understand if you consider an example in which this assumption is violated. Consider ice-cream sales over time. During a heat wave, sales will deviate from the model for a few days. The deviation (or error or residuals) will not be random or independent from each other for the time of the heatwave.

Detecting Violations:

- **Plotting Residuals**: Examine plots of residuals vs. fitted values. Patterns or trends in these plots can indicate violations.
- **Durbin-Watson Test**: Specifically used for detecting autocorrelation in the residuals from a regression analysis.

Let's look at an example in the Orthodont data (that comes with the `nlme` package) in which an X-ray parameter (`distance`) is explained by `age`. The `Orthodont` dataset contains measurements of the distance from the pituitary to the pterygomaxillary fissure in 27 children, measured at ages 8, 10, 12, and 14. There are both boys and girls in the study, and each child is measured multiple times.

In the plot below I illustrate how the same subjects are included in each of the age groups (and so are likely to be related to each other).

```
require(nlme)
```

```
Loading required package: nlme
```

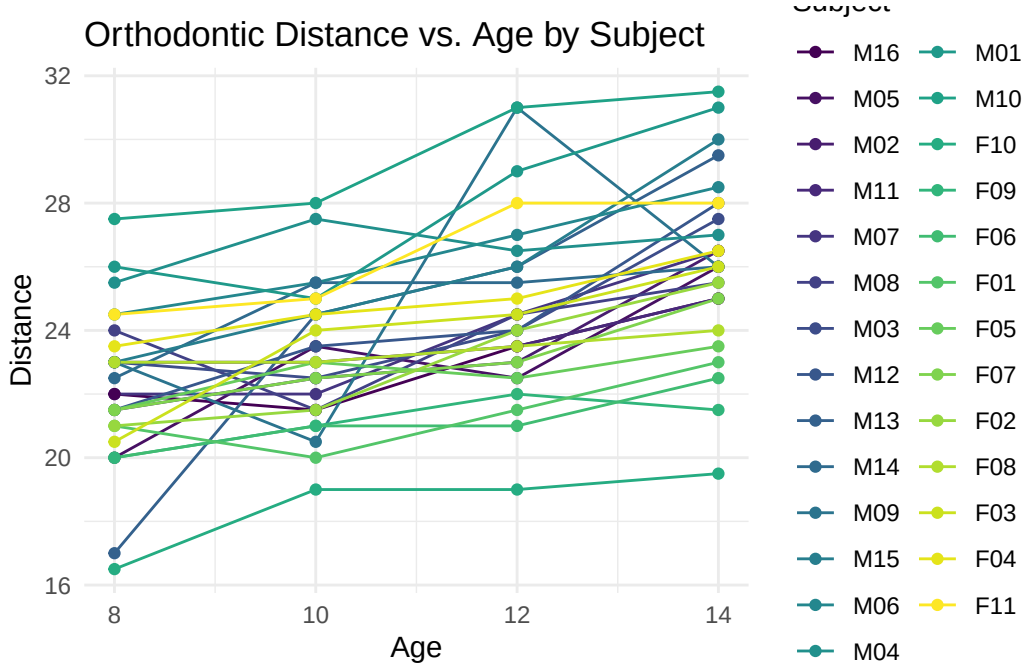
```
Attaching package: 'nlme'
```

```
The following object is masked from 'package:dplyr':

    collapse
```

```
ggplot(Orthodont, aes(x = age, y = distance, group = Subject, color = Subject)) +
  geom_line() +        # Add lines to connect points for each subject
  geom_point() +       # Add the actual data points
  theme_minimal() +    # Optional: a minimal theme for clarity
  labs(title = "Orthodontic Distance vs. Age by Subject",
```

```
  x = "Age",
  y = "Distance")
```



The assumption of independence of residuals is violated for the following reasons:

1. **Repeated Measures on Same Subjects**: Since each child is measured multiple times, the data points for each child are not independent of each other. The growth measurement at age 10, for instance, is likely to be correlated with the measurement at age 8 for the same child. This correlation between measurements violates the assumption of independent residuals.

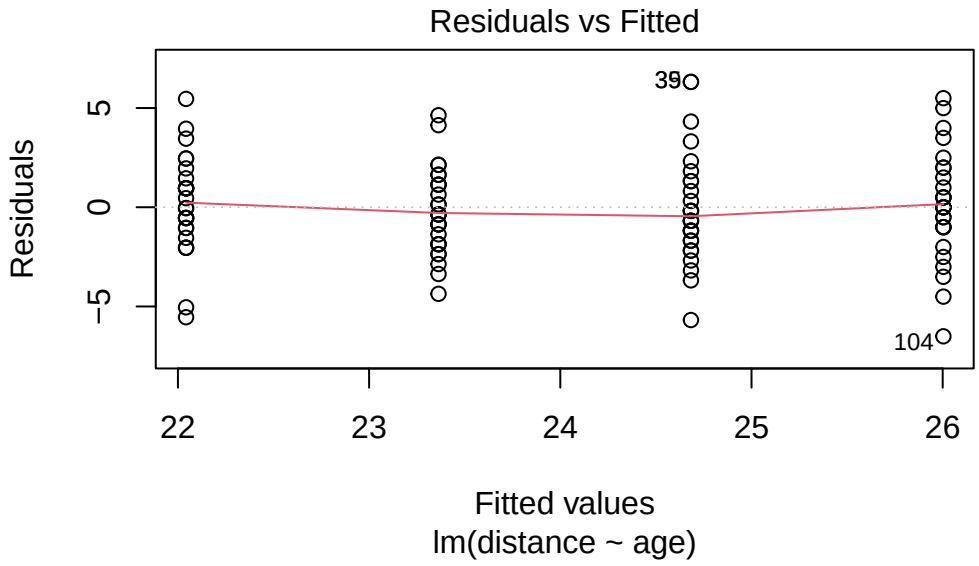
2. **Intra-Subject Correlation**: In longitudinal data like this, there's often a natural correlation within the subjects over time. A child who has a larger (or smaller) measurement than average at one time point is likely to have a larger (or smaller) measurement at another time point. This creates a scenario where residuals (differences between observed and predicted values) are not independent across time for the same individual.

3. **Growth Patterns and Individual Differences**: Each child may have unique growth patterns due to genetics, health, diet, and other factors. These individual differences can lead to correlated residuals within individuals, as the way one child's measurements change over time can be different from another child's.

We can try to visualize the relationship between the fitted values and residuals to determine if there is a patter but this isn't easy to interpret.

```r
model2 <- lm(distance ~ age, data = Orthodont)

plot(model2, which = 1)
```



A definitive method of determining independence is to us the Durbin Watson test (that comes with the `car` package).

```r
# install.packages("car")
library(car)
```

```
Loading required package: carData
```

```
Attaching package: 'car'
```

```
The following object is masked from 'package:dplyr':

    recode
```

```
The following object is masked from 'package:purrr':

    some
```

```
durbinWatsonTest(model2)
```

```
 lag Autocorrelation D-W Statistic p-value
   1       0.5399905     0.8912263       0
 Alternative hypothesis: rho != 0
```

A D-W statistic of close to 2 suggests that there is no autocorrelation, while a value below 2 suggests positive autocorrelation and above 2 suggests negative autocorrelation. In this case the value of 0.89 suggests positive autocorrelation.

Other considerations

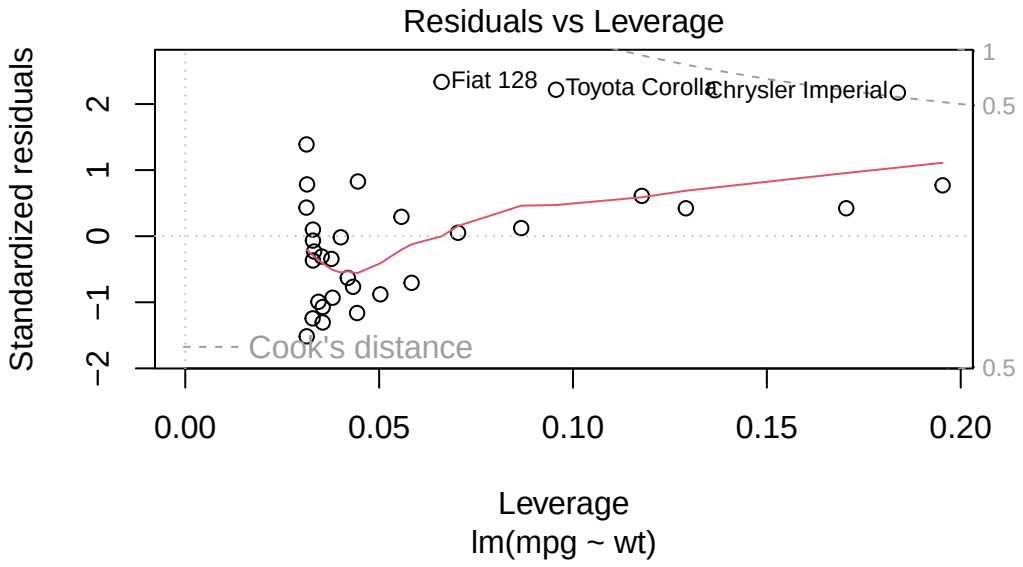
Outliers

When we think about outliers in the context of regression models, we're really talking about outliers with regard to the residual values. These are the values that tell us how well the observed values fit the model. Extreme values in the data itself may well fit the model very well with very small residual values.

Looking at the plot of residuals against fitted values (as seen above) you might see certain points that stand out as very far from the zero line (these are outliers).

Not all outliers will adversely affect your model (or have "leverage"). If they do they can be removed from the dataset before you fit the model.

By doing a **Leverage vs. Residual squared plot** you can identify outliers that will adversely influence your model. Creating the plot is surprisingly easy because it comes with a suit of diagnostic plots for regression models that are built into base R (we'll look more closely at these shortly). This is plot number 5.



Interpreting the Plot

1. **High Leverage Points**:
    - Points far to the right on the X-axis are high leverage points. These points can disproportionately influence the model because they are at extreme values of the explanatory variables.
2. **Large Residuals**:
    - Points far from the horizontal line at 0 on the Y-axis have large residuals. If they are beyond the -2 to 2 range, they are considered unusually large and might be outliers.
3. **Combination of Leverage and Residuals**:
    - Points with both high leverage and large residuals are of particular interest. They can strongly influence the regression line and potentially distort the analysis.
    - High leverage points with small residuals might not be problematic by themselves.
    - Points with large residuals but low leverage may indicate outliers that do not unduly influence the regression line.
4. **Patterns**:
    - Any pattern in the plot (like a systematic arrangement of points) might indicate non-linearity, heteroscedasticity, or other violations of regression assumptions.

Actionable Insights

- **Investigate Outliers**: Points with large standardized residuals should be investigated to determine if they are errors, special cases, or influential observations that need special attention.
- **Assess High Leverage Points**: High leverage points should be examined. Sometimes, they represent valuable information; other times, they might be data entry errors or anomalies.
- **Model Reevaluation**: If you find many high leverage points or patterns indicating model violations, consider reevaluating your model. This might include transforming variables, adding interaction terms, or using a different kind of regression model.

On this plot we can see that one observation is in the upper right quadrant. We can identify the outlier as follows:

```
high_leverage_point <- which.max(hatvalues(model))
print(row.names(mtcars)[high_leverage_point])
```

```
[1] "Lincoln Continental"
```

Now we could consider running the model with this data point excluded.

Collinearity,

This is relevant for multiple linear regression (where there are multiple predictors). It is the idea that the predictors should not be perfectly correlated with each other. Perfect multicollinearity means one predictor can be linearly predicted from the others.

Let's consider the mtcars data and the explanatory variables horse power (`hp`) and engine size (`disp`) as predictors of fuel efficiency (`mpg`). A quick correlation matrix is quite telling:

```
mtcars %>%
    select(mpg, disp, hp) %>%
    cor() %>%
    round(2)
```

```
       mpg  disp    hp
mpg   1.00 -0.85 -0.78
disp -0.85  1.00  0.79
hp   -0.78  0.79  1.00
```

It is clear that while both explanatory variables are correlated with `mpg` they are also highly correlated with each other. One might consider excluding one of them from the model because with respect to explaining `mpg` they are making an overlapping contribution. Imagine two people at a party, both telling the same story. You only need one. The other is creating noise.

While in general a highly collinear variable should be excluded from your model, the one notable exception is where it is an example of confounding. Think about the fact that ice-cream consumption is correlated with shark attack incidents. Clearly there is no causative relationship between the two. Instead, ice-cream consumption is associated with hotter weather. Hotter weather is associated with increase sea swimming and shark attacks. In the case of confounding variables, they must be included in the model to control for their effect. Confounding variables are usually identified through detailed subject and domain knowledge. Understanding the relationships between the variables is essential to build a good model.

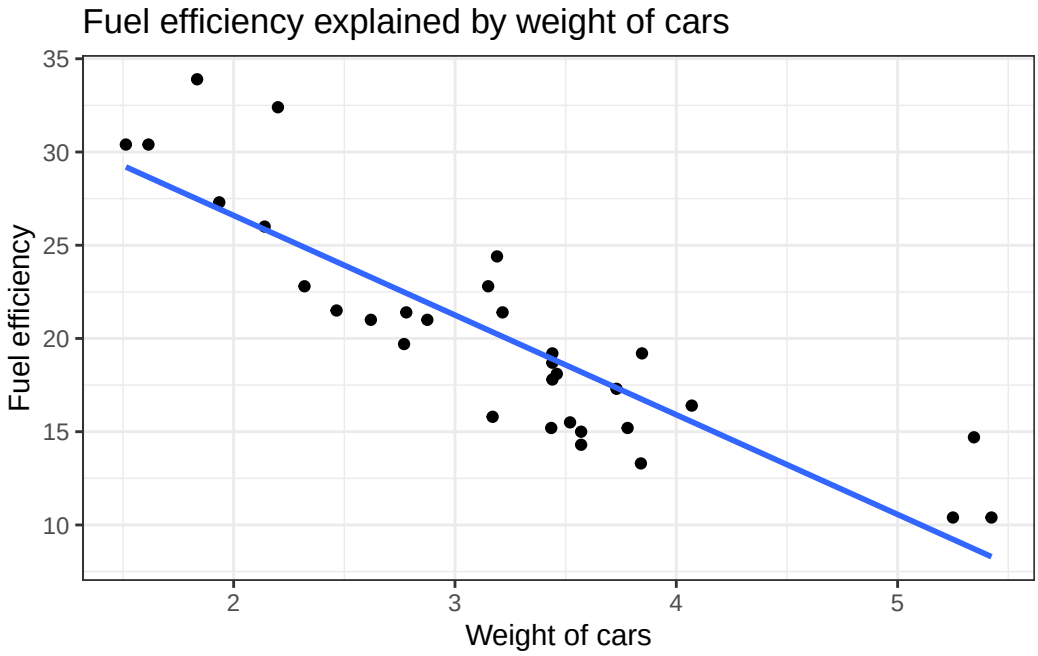
Effect modifiers and interaction

Where the relationship between an explanatory variable and an outcome variable changes depending on the value of a third variable, we call that third variable an effect modifier and say that there has been 'interaction'.

Let's start by looking at a simple linear model that considers the relationship between the weight (`wt`) of cars and the fuel efficiency (`mpg`) in the `mtcars` dataset.

```
mtcars %>%
    ggplot(aes(wt, mpg))+
    geom_point()+
    geom_smooth(method = lm, se = F)+
    theme_bw()+
    labs(title = "Fuel efficiency explained by weight of cars",
         x = "Weight of cars",
         y = "Fuel efficiency")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Now let's create a simple linear model to describe this:

```
lm(mpg~wt, data = mtcars) %>%
    summary()
```

```
Call:
lm(formula = mpg ~ wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
```

```
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

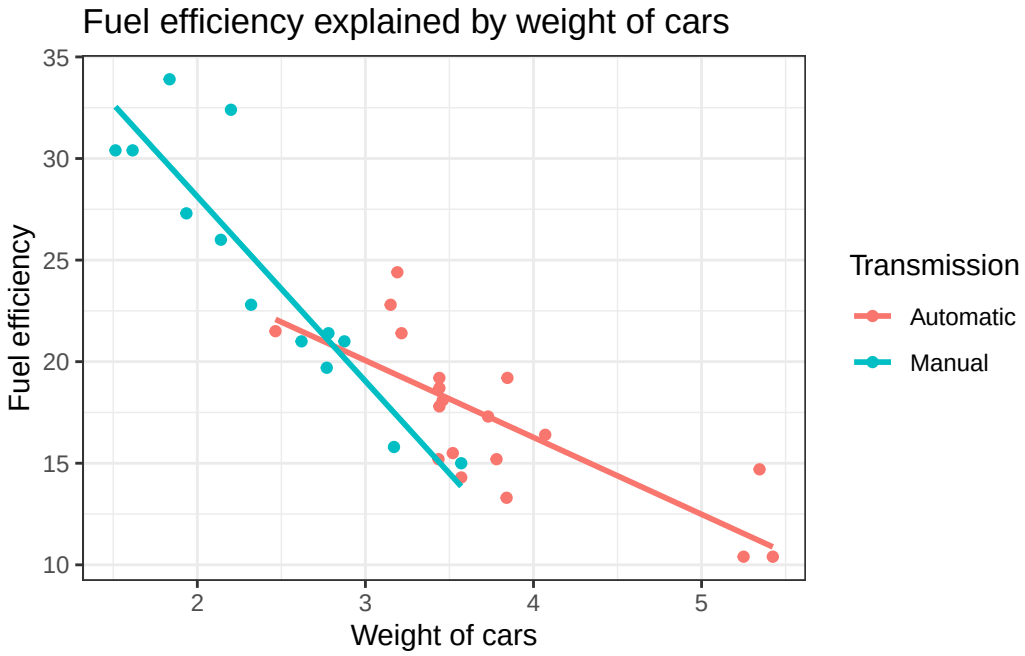
In this model, weight alone can explain 75% of fuel efficiency of cars. The slope coefficient is -5.34, so for a decrease in weight of 5,344 lbs, a car will gain 1 mile for every gallon of fuel used.

Now let's bring the number of cylinders in the car into the model.

Looking at the plot below, it is clear that the relationship between the weight of cars (`wt`) and the fuel efficiency (`mpg`) differs depending on whether the car transmission system (automatic or manual gears). Put another way, the slope coefficient changes as the number of cylinders changes; or the effect of weight on fuel efficiency depends on the transmission system of the car.

```
mtcars %>%
  mutate(am = factor(am, labels = c("Automatic", "Manual"))) %>%
  ggplot(aes(x = wt, y = mpg, color = am)) +
   geom_point() +
   geom_smooth(method = "lm", se = FALSE) +
   theme_bw()+
   labs(title = "Fuel efficiency explained by weight of cars",
         x = "Weight of cars",
         y = "Fuel efficiency",
         color = "Transmission")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Let's start off by simply adding cylinders to the model.

```
mtcars %>%
  mutate(am = factor(am, labels = c("Automatic", "Manual"))) %>%
  lm(mpg ~ wt + am, data = .) %>%
    summary()
```

```
Call:
lm(formula = mpg ~ wt + am, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5295 -2.3619 -0.1317  1.4025  6.8782

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 37.32155    3.05464  12.218 5.84e-13 ***
wt          -5.35281    0.78824  -6.791 1.87e-07 ***
amManual    -0.02362    1.54565  -0.015    0.988
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 3.098 on 29 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7358
F-statistic: 44.17 on 2 and 29 DF,  p-value: 1.579e-09
```

In the model above we've included transmission as an extra explanatory variable with no consideration for the fact that we believe that it effects the relationship between weight and fuel efficiency. Notice that transmission itself does not have a statistically significant slope coefficient. Note that in this mode, the Adjusted R-squared is actually lower (73.6%) than in the univariat analysis. The model is actually weaker.

To include transmission as a explanatory variable and include its interaction with weight we use `*` instead of `+`. In this case however we don't want to include transmission as an explanatory variable, but rather we'd like to only consider the contribution of the interaction - to do that we use `:` as follows:

```
mtcars %>%
  mutate(am = factor(am, labels = c("Automatic", "Manual"))) %>%
  lm(mpg ~ wt * am, data = .) %>%
  summary()
```

```
Call:
lm(formula = mpg ~ wt * am, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6004 -1.5446 -0.5325  0.9012  6.0909

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.4161     3.0201  10.402 4.00e-11 ***
wt           -3.7859     0.7856  -4.819 4.55e-05 ***
amManual     14.8784     4.2640   3.489  0.00162 **
wt:amManual  -5.2984     1.4447  -3.667  0.00102 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.591 on 28 degrees of freedom
Multiple R-squared:  0.833, Adjusted R-squared:  0.8151
F-statistic: 46.57 on 3 and 28 DF,  p-value: 5.209e-11
```

Let's look at the results of this analysis. Again, we won't comment on the intercept (as it is a meaningless number because the weight of a car can not be zero). The other coefficients are worth commenting on however:

- **wt -3.7859**: For each unit increase in weight (`wt`), `mpg` decreases by 3.79 miles per gallon.
- **amManual 14.8784**: Manual transmission cars are associated with an increase of 14.88 in `mpg` compared to automatics, assuming the car weight is kept constant.
- **wt:amManual -5.2984**: The interaction term indicates that for manual transmission cars, the negative impact of weight on `mpg` is more pronounced by an additional -5.3 per unit increase in weight compared to automatic transmission cars.

All three coefficients are statistically significant and the model now allows us to explain 82% of the change in `mpg` (the adjusted R-squared is 0.815)

Now lets consider interaction with a numeric variable. In this case we'll look at horse power (`hp`):

```
mtcars %>%
    lm(mpg ~ wt*hp, data = .) %>%
    summary()
```

```
Call:
lm(formula = mpg ~ wt * hp, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0632 -1.6491 -0.7362  1.4211  4.5513

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49.80842    3.60516  13.816 5.01e-14 ***
wt          -8.21662    1.26971  -6.471 5.20e-07 ***
hp          -0.12010    0.02470  -4.863 4.04e-05 ***
wt:hp        0.02785    0.00742   3.753 0.000811 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.153 on 28 degrees of freedom
Multiple R-squared:  0.8848,    Adjusted R-squared:  0.8724
F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13
```

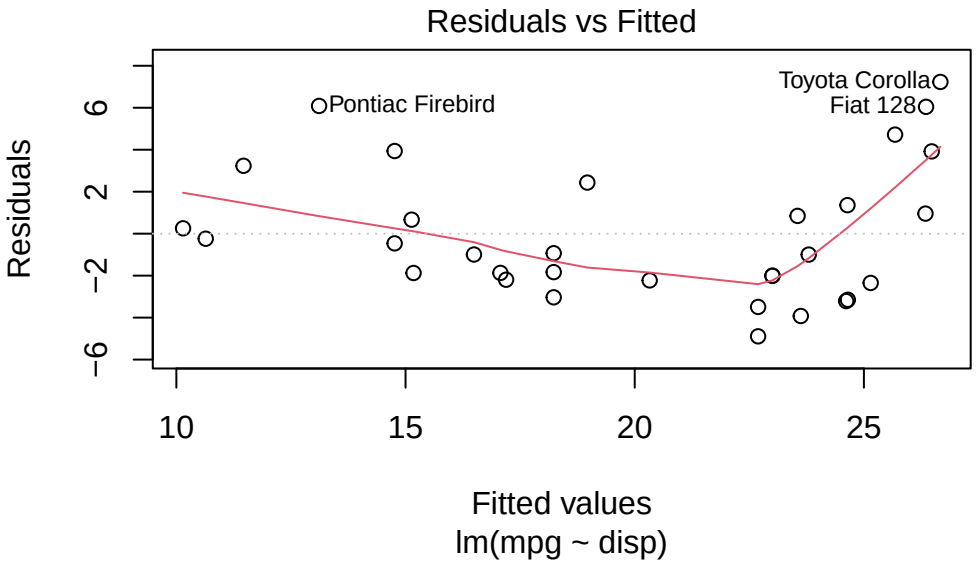
First note that the interaction between `wt` and `hp` is statistically significant and the resultant model is now able to explain 87% of the change in fuel efficiency (`mpg`). Looking at the coefficients:

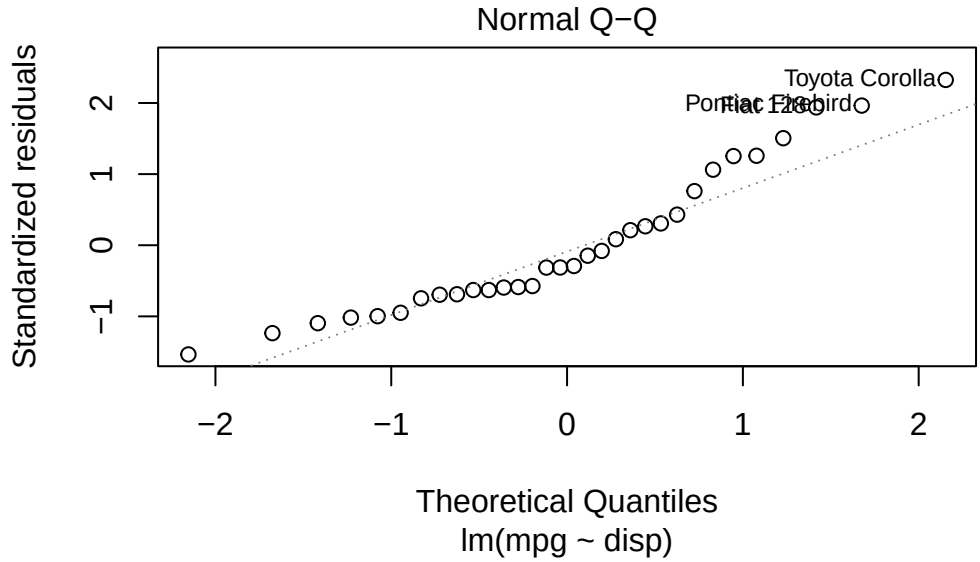
- **wt (-8.21662)**: The coefficient for `wt` is -8.22, indicating that for each unit increase in weight, holding horsepower constant, the `mpg` is expected to decrease by about 8.22 units. This is a substantial negative impact, suggesting that heavier cars generally have lower fuel efficiency.
- **hp (-0.12010)**: The coefficient for `hp` is -0.12, showing that for each unit increase in horsepower, with the weight held constant, the `mpg` decreases by 0.12 units. This suggests that higher horsepower (which often correlates with faster, more powerful cars) also tends to decrease fuel efficiency, but the effect is less pronounced than the effect of weight.
- **wt:hp (0.02785)**: The interaction term coefficient is positive, indicating that the negative impact of weight on `mpg` becomes less negative (or more positive) at higher horsepower values. In other words, for heavier cars, the penalty on fuel efficiency is lower per additional horsepower than it is for lighter cars. This interaction effect is significant and important to consider in the overall model.

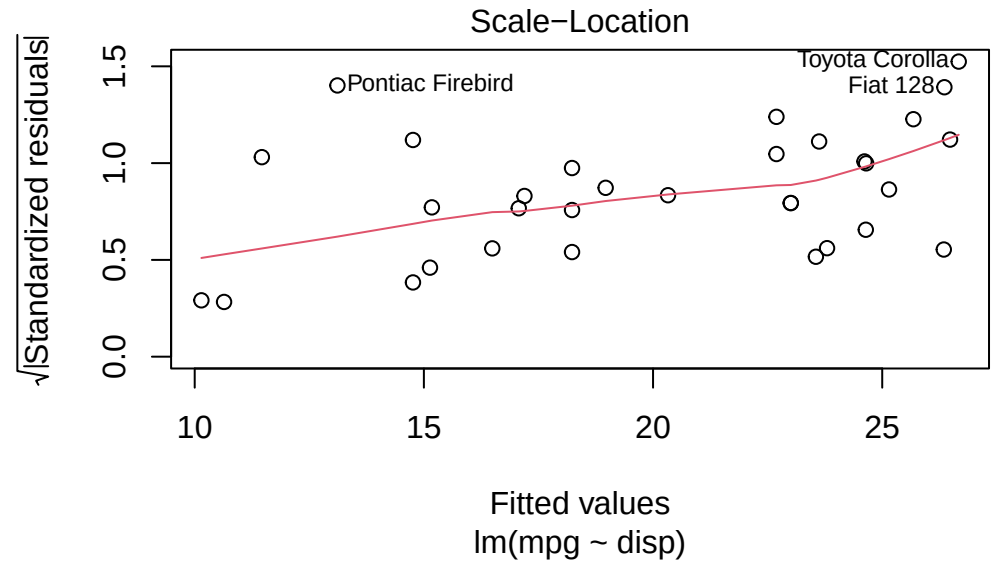
Generating diagnostics plots

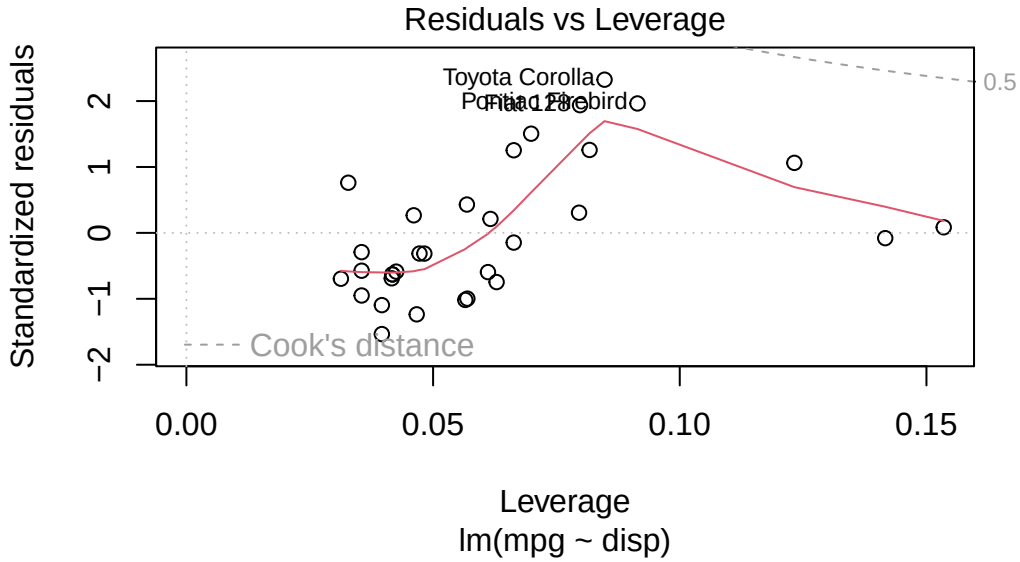
To make things easy, R provides a simple way to get the diagnostic plots that you need to look at the various assumptions that you've made when doing linear regression. Let's look at how:

```
model3 <- lm(mpg ~ disp, data = mtcars)
plot(model3)
```


Residuals vs Fitted
Residuals
Pontiac Firebird
Toyota Corolla
Fiat 128
6
2
−2
−6
10
15
20
25
Fitted values
lm(mpg ~ disp)


Normal Q−Q
Standardized residuals
Toyota Corolla
Pontiac Firebird
Fiat 128
2
1
0
−1
−2
−1
0
1
2
Theoretical Quantiles
lm(mpg ~ disp)





These plots should be used in conjunction with statistical tests used to assess the assumptions of your model. The Scale-Location plot is less easy to interpret because the y-axis represents normalised and square rooted residuals (its difficult to have an intuitive sense of what "good" should look like. By contrast the fitter values vs residuals is much easier to understand and if combined with statsitical testing should be sufficient to examine the assumptions of your residuals.