# Practical 4: Finding Relationships in R

An Introduction to Spatial Data Analysis and Visualisation in R - Guy Lansley & James Cheshire (2016)

This practical is intended to introduce you to some of the most commonly used means of statistically identifying and measuring bivariate and multivariate relationships in R. Data for the practical can be downloaded from the **Introduction to Spatial Data Analysis and Visualisation in R** (https://data.cdrc.ac.uk/tutorial/an-introduction-to-spatial-data-analysis-and-visualisation-in-r) homepage.

In this tutorial we will:

- Run a Pearson's correlation test
- Run a Spearman's correlation test
- Run a linear regression model

First, we must set the working directory and load the practical data. Remember to change the working directory to the correct file path on your computer.

```
#Set the working directory. Remember to alter the file path below.
setwd("C:/Users/Guy/Documents/Teaching/CDRC/Practicals")

#Load the data. You may need to alter the file directory
Census.Data <-read.csv("practical_data.csv")
```

# Bivariate correlations

One common means of identifying and measuring the relationship between two variables is a correlation (http://www.statsref.com/HTML/?correlation.html). In R this can simply be done by using the `cor()` function and inputting two variables within the parameters - i.e:

```
# Runs a Pearson's correlation
cor(Census.Data$Unemployed, Census.Data$Qualification)
```

This will return a Pearson's correlation (http://www.statsref.com/HTML/?pearson_product_moment_correla.html) coefficient(*r*). A Pearson's (or Product Moment Correlation) coefficient (http://www.statsref.com/HTML/?pearson_product_moment_correla.html) is a measure of linear association between two variables. Greater values represent a stronger relationship between the pair. *1* represents a perfect positive relationship, *0* represents no linear correlation and *-1* represents a perfect negative relationship.

In R, a better option is to use `cor.test()` as this also reports significance statistics. If a test is not statistically significant, its results cannot be regarded as reliable. Here is an example below.

```
# Runs a Pearson's correlation
cor.test(Census.Data$Unemployed, Census.Data$Qualification)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Census.Data$Unemployed and Census.Data$Qualification
## t = -21.85, df = 747, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6662641 -0.5786800
## sample estimates:
##       cor
## -0.624431
```

The final value is the Pearson's correlation coefficient. A score of -0.62 identifies that there is a negative relationship between the unemployment and qualification variables. From the model, we also get the 95% confidence intervals. Confidence intervals display the range of values of which there is a defined probability that the coefficient falls within. The output also returns the result of the t-test. We can use this to determine if the results were statistically significant.

A Pearson's correlation is only suitable when the relationship between the two variables is linear. It is not sensitive to relationships that are non-linear. In these circumstances, it is worth using Spearman's rank correlation (http://www.statsref.com/HTML/?rank_correlation.html). This statistic is obtained by simply replacing the observations by their rank within their sample and computing the correlation, which means it is also suitable for large-scale ordinal variables.

```
# Runs a Spearman's correlation
cor.test(Census.Data$Unemployed, Census.Data$Qualification, method="spearman")
```

```
## Warning in cor.test.default(Census.Data$Unemployed, Census.Data
## $Qualification, : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  Census.Data$Unemployed and Census.Data$Qualification
## S = 113730000, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6240406
```

Does your conclusion about the relationship between these two variables change when using a Spearman's correlation compared with a Pearson's correlation?

It is also possible to produce a correlation pair-wise matrix in R. This will display a correlation coefficient for every possible pairing of variables in the data. To do this we need to first format the data to get rid of the ID column as it will not work in a correlation. We only want to include the variables for our correlation matrix.

```
# creates a data1 object which does not include the 1st column from the original d
ata
data1 <- Census.Data[,2:5]
```

Then with our new data1 object, we can create a new matrix.

```
# creates correlation matrix
cor(data1)
```

```
##                White_British Low_Occupancy Unemployed Qualification
## White_British      1.0000000    -0.6006639 -0.3984454     0.4992319
## Low_Occupancy     -0.6006639     1.0000000  0.6408021    -0.7347354
## Unemployed        -0.3984454     0.6408021  1.0000000    -0.6244310
## Qualification      0.4992319    -0.7347354 -0.6244310     1.0000000
```

Remember coefficients of 1 will always be produced between identical variables as they display the same patterns.

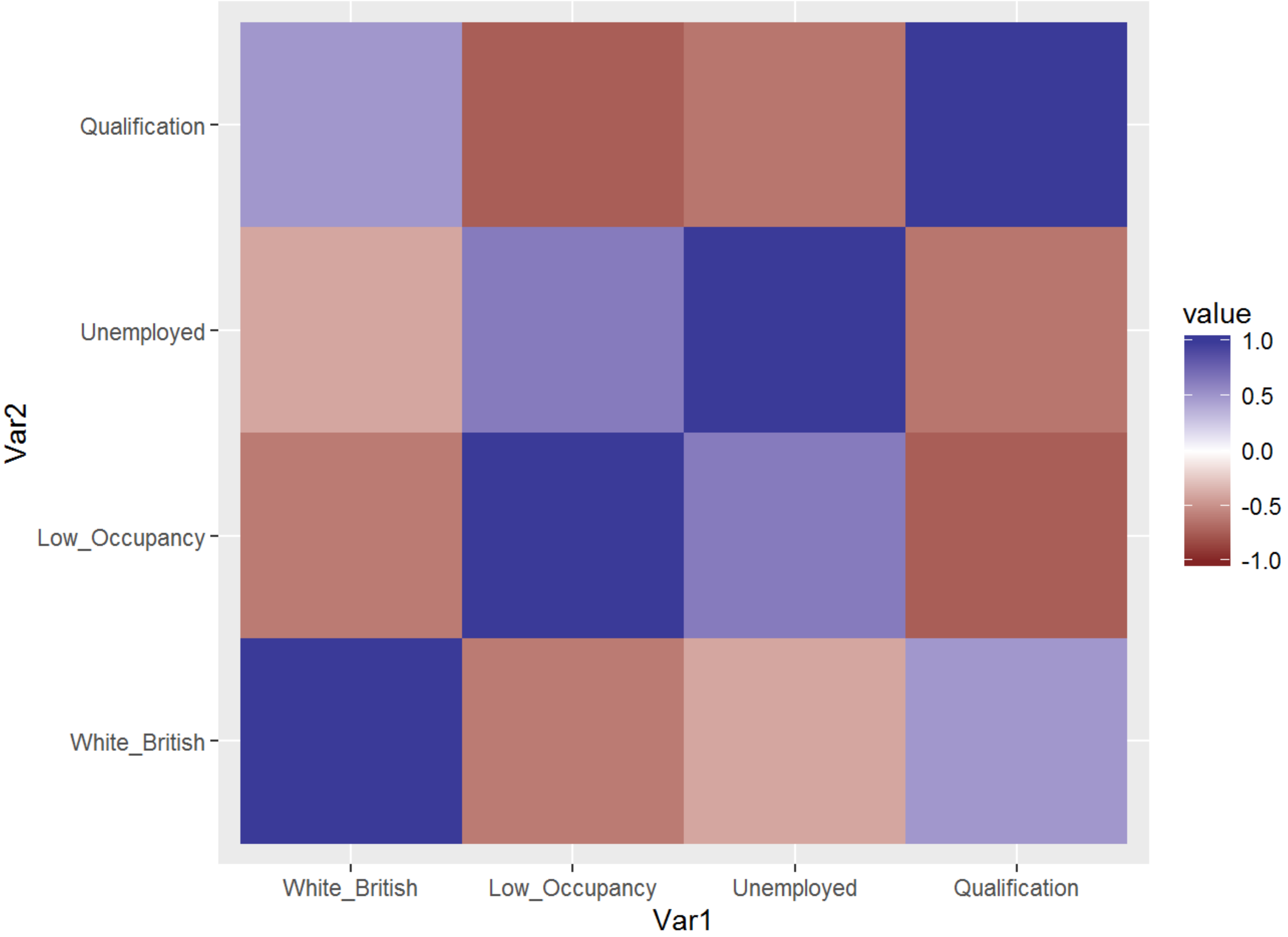
We can use the `round()` function to round our results to 2 decimal places.

```
# creates correlation matrix
round(cor(data1),2)
```

```
##                White_British Low_Occupancy Unemployed Qualification
## White_British            1.0         -0.60      -0.40          0.50
## Low_Occupancy           -0.6          1.00       0.64         -0.73
## Unemployed              -0.4          0.64       1.00         -0.62
## Qualification            0.5         -0.73      -0.62          1.00
```

We can use the `qplot()` function from the ggplot2 package to create a heat map of this correlation matrix. The code to do this is written below.

```
library(ggplot2) # should already be opened from the previous stage
library(reshape2)
qplot(x=Var1, y=Var2, data=melt(cor(data1, use="p")), fill=value, geom="tile") +
   scale_fill_gradient2(limits=c(-1, 1))
```



Heat maps like this can be a simple and effective means of conveying lots of information in one graphic. In this case, the correlation matrix is already quite small so obscuring the numerical values with colours is not necessary. But if you have very large matrices of say 50+ cells this could be a useful technique.

# Regression analysis

A simple linear regression (http://www.statsref.com/HTML/?regression.html) plots a single straight line of predicted values as the model for a relationship. It is a simplification of the real world and its processes, that assumes that there is approximately a linear relationship between x and y.

Another way of thinking about this line is as the best possible summary of the cloud of points that are represented in the scatterplot (if we can assume that a straight line would do a good job doing this). If I were to tell you to draw a straight line that best represents this pattern of points the regression line would be the one that best does it (if certain assumptions are met). The linear model then is a model that takes the form of the equation of a straight line through the data. The line does not go through all the points.

In order to draw a regression line we need to know two things:

1. We need to know where the line begins - the value of y (our dependent variable) when x (our independent variable) is 0 - so that we have a point from which to start drawing the line. The technical name for this point is the intercept or the constant.
2. And we need to know what is the slope of that line.

If you recall from school algebra (and you may not), the equation for any straight line is: y = mx + b. In statistics, we use a slightly different notation, although the equation remains the same: y = b0 + b1x.

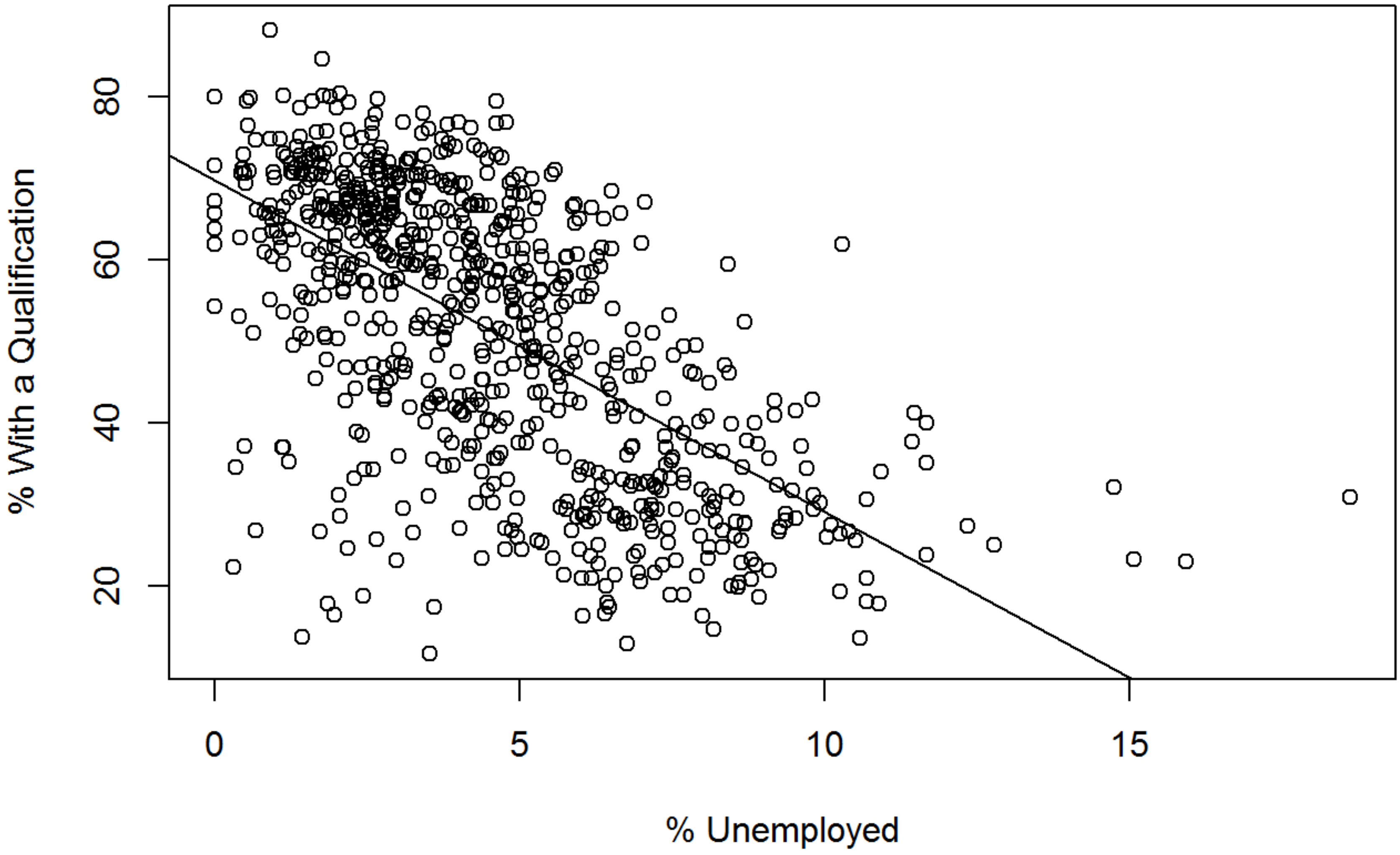
We need the origin of the line (b0) and the slope of the line (b1). How does R get the intercept and the slope of the regression line? How does R know where to draw this line? We need to estimate these parameters (or coefficients) from the data. For linear regression models (like the one we cover here) R tries to minimise the distance from every point in the scatterplot to the regression line using a method called least squares estimation.

In order to fit the model we use the `lm()` function using the formula specification (y ~ x). Typically you want to store your regression model in a “variable”, let’s call it model_1:

```
model_1 <- lm(Census.Data$Qualification~ Census.Data$Unemployed)
```

First, let’s add the regression line from the model to a scatter plot by using the `abline()` function (as we did in the previous practical. Notice that the model orders the x and y the other way round.

```
plot(Census.Data$Unemployed, Census.Data$Qualification, xlab="% Unemployed", ylab=
"% With a Qualification") + abline (model_1)
```



```
## numeric(0)
```

If you want to simply see the basic results from running the model you can use the `summary()` function.

```
summary(model_1)
```

```
##
## Call:
## lm(formula = Census.Data$Qualification ~ Census.Data$Unemployed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.172  -9.635   2.339   9.512  36.887
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             69.7740     0.9743   71.61   <2e-16 ***
## Census.Data$Unemployed  -4.0672     0.1861  -21.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.53 on 747 degrees of freedom
## Multiple R-squared:  0.3899, Adjusted R-squared:  0.3891
## F-statistic: 477.4 on 1 and 747 DF,  p-value: < 2.2e-16
```

For now, just focus on the numbers in the “Estimate” column. The value of 69.78 estimated for the intercept is the “predicted” value for y when x equals zero - it is possible to interpret this from observing the scatter plot we just made. This is the predicted value of the percentage of people with degrees when the percentage of people who are unemployed is zero.

We then need the b1 regression coefficient for our independent variable (Unemployed), the value that will shape the slope in this scenario. This value is -4.0672. This estimated regression coefficient for our independent variable has a convenient interpretation. When the value is positive, it tells us that for every one unit increase in X there is a b1 increase on y. If the coefficient is negative then it represents a decrease on y. Here, we can read it as “for every one unit increase in the percentage of people who are unemployed, there is a -4.0672 unit decrease in the percentage of people with a degree.”

Knowing these two parameters not only allows us to draw the line, we can also solve y for any given value of x. If the percentage of people who are unemployed in a given area is 15%, we can simply go back to our regression line equation and insert the estimated parameters:

*y = b0 + b1x or y = 69.78 + (-4.0672 x 15)*

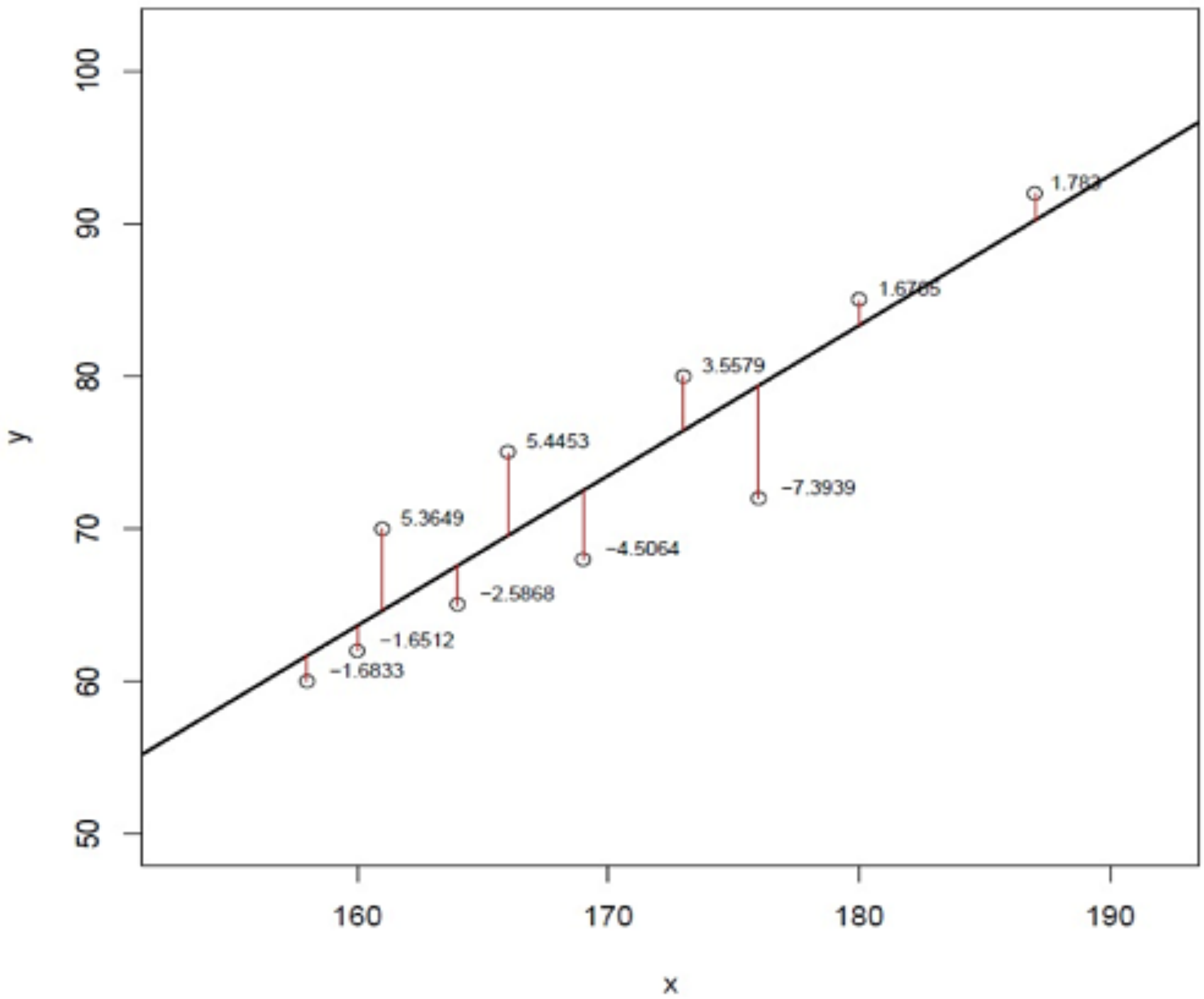
If you don’t want to do the calculation yourself, you can use the predict function:

```
predict(model_1, data.frame(Unemployed = c(15)))
```

Of course this model is simplification of reality and only considers the influence of one variable.

## R squared

In the output above we saw there was something called the residuals. The residuals are the differences between the observed values of y for each case minus the predicted or expected value of y, in other words, the distances between each point in the dataset and the regression line (see the visual example below).

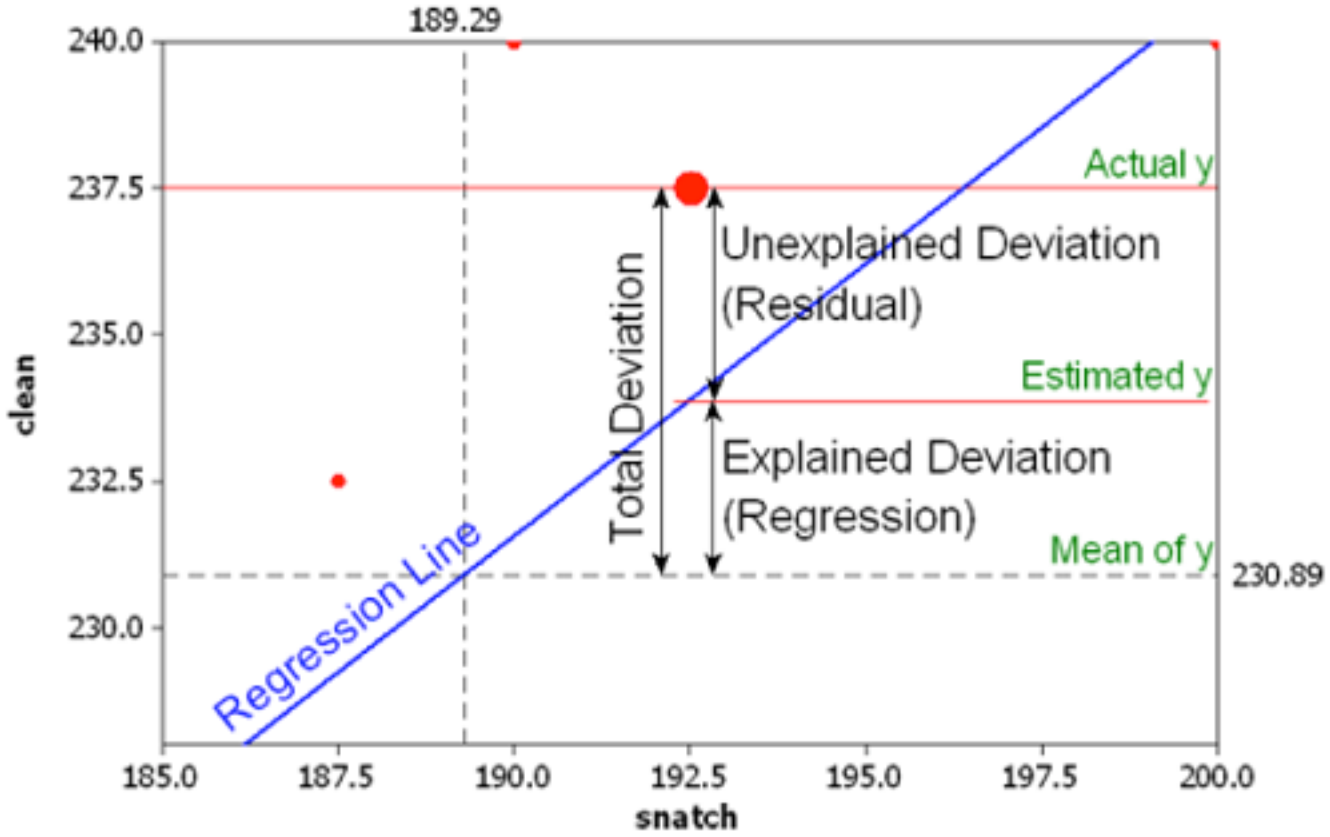


We see here points indicating the observed values (the data points in our sample), red lines indicating their distance to the regression lines (the residuals), and numerical values indicating the distance from each of the points to the regression line. Least square estimation essentially aims to reduce the squared average of all these distances: that's how it draws the line.

We have residuals because our line is not a perfect representation of the cloud of points. You cannot predict perfectly what the value of y is for every area just by looking ONLY at the value of x. There are other things that may influence the values of y which are not being taken into account by our model. There are other things that surely matter in terms of understanding of the relationship between health and levels of education. And then, of course, we have measurement error and other forms of noise.

We can re-write our equation like this if we want to represent each value of y (rather than the predicted value of y) then: *y = b0 + b1x + error*

The residuals capture how much variation is unexplained, how much we still have to learn if we want to understand variation in y. A good model tries to maximise explained variation and reduce the magnitude of the residuals. We can use information from the residuals to produce a measure of effect size, of how good our model is in predicting variation in our dependent variables. If we did not have any information about x our best bet for y would be the mean of y. The regression line aims to improve that prediction. By knowing the values of x we can build a regression line that aims to get us closer to the actual values of y (look at the Figure below).



The distance between the mean (our best guess without any other piece of information) and the observed value of y is what we call the total variation. The residual is the difference between our predicted value of y and the observed value of y. This is what we cannot explain (i.e, variation in y that is unexplained). The difference between the mean value of y and the expected value of y (the value given by our regression line) is how much better we are doing with our prediction by using information about x. How much closer the regression line gets us to the observed values. We can then contrast these two different sources of variation (explained and unexplained) to produce a single measure of how good our model is. The formula is as follows:

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

All this formula is doing is taking a ratio of the explained variation (the squared differences between the regression line and the mean of Y for each observation) by the total variation (the squared differences of the observed values of Y for each observation from the mean of Y). This gives us a measure of the percentage of variation in Y that is “explained” by X.

We can take this value as a measure of the strength of our model. If you look at the R output you will see that the R2 for our model was 0.3899 (look at the multiple R2 value in the output). We can say that our model explains about 40% of the variance in the percentage of people with a degree in our study area.

Knowing how to interpret this is important. R2 ranges from 0 to 1. The greater it is, the more powerful our model is, the more explaining we are doing, and the better we are able to account for variation in our outcome Y with our input. In other words, the stronger the relationship is between Y and X. As with all the other measures of effect size, interpretation is a matter of judgement. You are advised to see what other researchers report in relation to the particular outcome that you may be exploring. We can use the R2 to compare against other models we might fit to see which is most powerful.

## Inference with regression

In real world applications, we have access to a set of observations from which we can compute the least squares line, but the population regression line is unobserved. So our regression line is one of many that could be estimated. A different set of Output Areas would produce a different regression line. If we estimate b0 and b1 from a particular sample, then our estimates won’t be exactly equal to b0 and b1 in the population. But if we could average the estimates obtained over a very large number of data sets, the average of these estimates would equal the coefficients of the regression line in the population.

In the same way that we can compute the standard error when estimating the mean, we can compute standard errors for the regression coefficients to quantify our uncertainty about these estimates. These standard errors can, in turn, be used to produce confidence intervals. This would require us to assume that the residuals are normally distributed. For a simple regression model, you are assuming that the values of y are approximately normally distributed for each level of x:

You can also then perform a standard hypothesis test on the coefficients. As we saw before when summarising the model, R will compute the standard errors and a t-test for each of the coefficients.

In our example, we can see that the coefficient for our predictor here is statistically significant, as represented by the p-value. Notice that the t-statistics and p-value are the same as the correlation coefficient.

```
summary(model_1)
```

```
## 
## Call:
## lm(formula = Census.Data$Qualification ~ Census.Data$Unemployed)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -50.172  -9.635   2.339   9.512  36.887 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             69.7740     0.9743   71.61   <2e-16 ***
## Census.Data$Unemployed  -4.0672     0.1861  -21.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.53 on 747 degrees of freedom
## Multiple R-squared:  0.3899, Adjusted R-squared:  0.3891 
## F-statistic: 477.4 on 1 and 747 DF,  p-value: < 2.2e-16
```

We can also obtain confidence intervals for the estimated coefficients using the confint() function. The below example will produce a 95% confidence interval for the model. The 95% confidence interval defines a range of values that you can be 95% certain contains the mean slope of the regression line.

```
confint(model_1, level= 0.95)
```

```
##                            2.5 %    97.5 %
## (Intercept)            67.861262 71.686689
## Census.Data$Unemployed -4.432593 -3.701747
```

## Mutliple regression

So we have seen our models with just one predictor or explanatory variable. We can build ‘better’ models by increasing the number of predictors. In our case, we can also add another variable into the model for predicting the number of people with degree level qualifications. We have seen from the plots above that there are clearly fewer people living in deprivation in areas where more people have a degree, so let's see if it helps us make better predictions.

Another reason why it is important to think about additional variables in your model is to control for spurious correlations (although here you may also want to use your common sense when selecting your variables!). You have heard that correlation does not equal causation. Just because two things are associated we cannot assume that one is the cause for the other. Typically we see how the pilots switch the “secure the belt” sign on when there is turbulence during a flight. These two things are associated, they tend to come together. But the pilots are not causing the turbulence by pressing a switch! The world is full of spurious correlations, associations between two variables that should not be taken too seriously.

It's not an exaggeration to say that most quantitative explanatory research is about trying to control for the presence of confounders - variables that may explain away observed associations. Think about any social science question: Are married people less prone to depression? Or is it that people that get married are different from those that don't (and are there pre-existing differences that are associated with less depression)? Are ethnic minorities more likely to vote for centre-left political parties? Or, is it that there are other factors (e.g. socioeconomic status, area of residence, sector of employment) that, once controlled, would mean there is no ethnic group difference in voting?

Multiple regression is one way of checking the relevance of competing explanations. You could, for example, set up a model where you try to predict voting behaviour with an indicator of ethnicity and an indicator of structural disadvantage. If after controlling for structural disadvantage, you see that the regression coefficient for ethnicity is still significant you may be onto something, particularly if the estimated effect is still large. If, on the other hand, the t-test for the regression coefficient of your ethnicity variable is no longer significant, then you may be tempted to think that structural disadvantage is a confounder for vote selection.

It could not be any easier to fit a multiple regression model. You simply modify the formula in the `lm()` function by adding terms for the additional inputs. Here the 2nd predictor (or independent) variable is the % of the white British population.

```
model_2 <- lm(Census.Data$Qualification~ Census.Data$Unemployed + Census.Data$White_British)

summary(model_2)
```

```
##
## Call:
## lm(formula = Census.Data$Qualification ~ Census.Data$Unemployed +
##     Census.Data$White_British)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.311  -8.014   1.006   8.958  38.046
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               47.86697    2.33574   20.49   <2e-16 ***
## Census.Data$Unemployed    -3.29459    0.19027  -17.32   <2e-16 ***
## Census.Data$White_British  0.41092    0.04032   10.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.69 on 746 degrees of freedom
## Multiple R-squared:  0.4645, Adjusted R-squared:  0.463
## F-statistic: 323.5 on 2 and 746 DF,  p-value: < 2.2e-16
```

Now we can consider the influence of multiple variables. Also, you will notice that the R2 value has improved slightly compared to the first model. Try testing various different combinations of variables. Which model is most efficient at representing the distribution of our qualifications variable across the study area?

---

The rest of the online tutorials in this series can be found at: https://data.cdrc.ac.uk/tutorial/an-introduction-to-spatial-data-analysis-and-visualisation-in-r (https://data.cdrc.ac.uk/tutorial/an-introduction-to-spatial-data-analysis-and-visualisation-in-r)