GEOG0013 Statistical Analysis Worksheet 3: Relationships between variables

Overview

Previously you explored the distribution of values on one variable at a time. However, as geographers we are often especially keen to understand the relationship between two or more variables (bivariate and multivariate analysis respectively). If we find that there is a relationship between two variables, we then often want to quantify *the magnitude/strength* and *direction* of it. Do high values on X generally pair up with high values on Y, or does Y tend to increase as X falls? If they are associated, then how strongly do the two variables tend to move together?

You are now going to learn how R can be used to explore relationships between variables. We are going to work with the Barc_neighbs data frame you prepared in the previous session. Your task will be to examine the relationship between average rent (Y) and unemployment rate (X) across the city's neighbourhoods. Take a moment before you begin to jot down your hypothesis about what you expect to find. Will the two variables be related, and if so, how? Why do you expect to find this pattern?

The first part of the session shows you how to produce *correlation coefficients* and draw scatter plots to describe variable relationships. You will then create *regression models* to examine these relationships. Regression provides a flexible modelling framework that allows you to "explain" or "predict" a given outcome (Y- the dependent or outcome variable) - as a function of a number of what are variously called independent, explanatory, or predictive variables (denoted by X1, X2, X3, etc). Although there are a very large family of regression techniques, if you are interested in modelling variations in a single continuous y variable as a function of a set of x variables then linear regression or ordinary least squares regression (OLS) is most often used.

Worksheet aims

1. Learn how to produce scatter plots
2. Compute a correlation coefficient and test it for 'statistical significance'
3. Create and interpret a linear regression model
4. Fit a multiple regression model with two X variables

Scatter plots

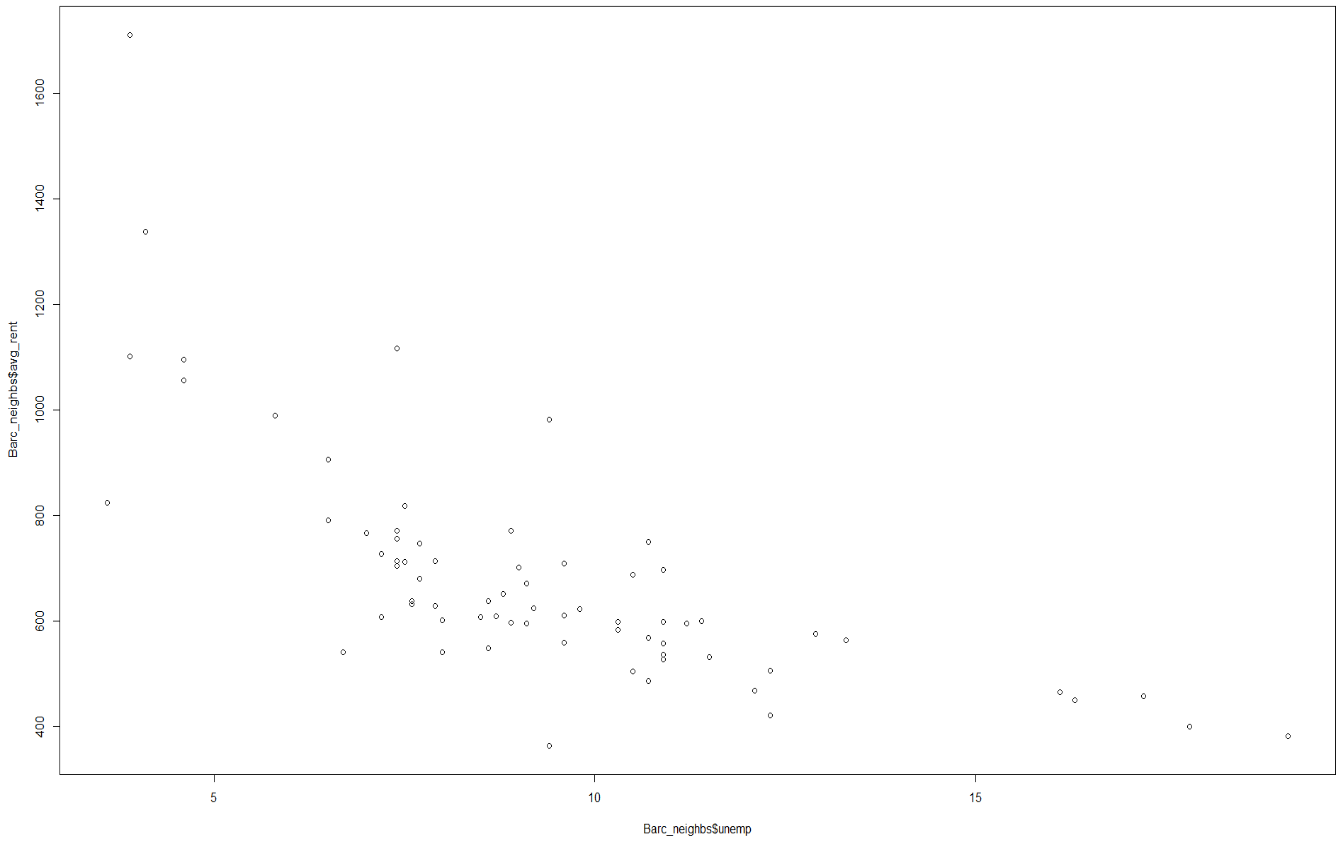
You should have the data frame object you will need (Barc_neighbs) in your R session from before the break. If for some reason it is not loaded in R (e.g. your computer crashed) then you may need to set the working directory again as was explained in the last session. When you have done that you can then reload the Barc_neighbs data using **read.csv**(). You do not need to run the code snippet below if Barc_neighbs is already loaded into your R session!

```
Barc_neighbs <- read.csv("Barcelona.csv")
```

Perhaps the best place to begin exploring the relationship between the unemp (% unemployed) and avg_rent (average rent in €) variables is a scatter plot. The standard installation of R has base graphic functionality built in to produce very simple plots through the **plot()** function:

```
#left of the comma is the x-axis, right is the y-axis. Also note again how we are
using the $ symbol to select the columns of the data frame we want.

plot(Barc_neighbs$unemp, Barc_neighbs$avg_rent)
```



You should see this simple scatterplot. Can you interpret the relationship between unemployment (X) and rent levels (Y) in Barcelona? Are the variables positively correlated (as X increases so does Y), negatively correlated (Y falls as X increases) or is there no relationship (points are randomly spread out so Y and X are unrelated)? How strong does this relationship appear to be? A strong correlation is when the relationship between X and Y resembles a straight line, whereas a weak correlation has more dispersion around a line of best fit.

The plot command offers a huge number of options for customisation. You can see them using the ?plot help pages and also the ?par help pages (par in this case is short for parameters). There are some examples below (note how the parameters come after the x and y columns).

```
#Add axis labels
plot(Barc_neighbs$unemp, Barc_neighbs$avg_rent, ylab="Average rent (€)", xlab="%
unemployed in neighbourhood")
```

```
#Add a title (main =), change point colour (col =), change point size (cex =)
plot(Barc_neighbs$unemp, Barc_neighbs$avg_rent, main="Relationship between
unemployment rate and rents in Barcelona's neighbourhoods (2015)", col="blue",
cex=2)
```

```
#Add a title, change point colour, change point symbol (pch =)
plot(Barc_neighbs$unemp, Barc_neighbs$avg_rent, main="Relationship between
unemployment rate and rents in Barcelona's neighbourhoods (2015)", col="red",
pch=22)
```

On your own, try adding x and y labels by adding xlab=" " and ylab=" " options to the last segment of code (place the labels between the quotes). Remember that spreading your code across several lines of your script is usually better than producing a very wide script. To run code spanning several lines just highlight the block of code and press Run or cntrl-enter.

Note: if you would like to save your plots for use in documents you can click on the "Export" button on the plot window to save or copy to clipboard.

Correlation coefficients

The plots you just ran visualise the relationship between unemployment and rents across Barcelona's neighbourhoods. However, it is valuable to also *quantify* the strength and direction of bivariate associations with single numbers.

A great advantage of this approach is that we can run statistical tests to examine if an association is significantly different from 0 (i.e. it is extremely unlikely to be found in your particular sample of data if actually there is no real relationship in the wider population from which the sample was drawn – this 'no actual relationship' scenario is what we term our null hypothesis and it is what we are testing). By convention, we often conclude that a relationship is 'statistically significant' if R tells us that it has a p-value *below* 0.05. In simple terms, a low p-value like this means that with your particular sample of data, you would be very unlucky to find a correlation at least as strong as the one you found simply due to chance variation across samples if actually there is no relationship between x and y in the wider population you took your sample from (and which you don't know about).

The Pearson's Correlation Coefficient (r) measures the strength of a linear (straight-line) relationship between two variables. The coefficient takes a value between -1 and +1, where -1 indicate a perfect negative relationship, +1 indicates a perfect positive relationship and 0 indicates no relationship at all. In R the function **cor.test()** can be used to compute a correlation coefficient between two or more variables, in this case unemployment and rents:

```r
cor.test(Barc_neighbs$unemp, Barc_neighbs$avg_rent)
```

The results are printed to the console as shown below. The value of r (the Pearson's correlation coefficient) is highlighted in yellow and its p-value is highlighted in red. Note that the p-value is shown in a form of scientific notation called e-notation because it is miniscule (https://en.wikipedia.org/wiki/Scientific_notation).

```
Pearson's product-moment correlation

data:  Barc_neighbs$unemp and Barc_neighbs$avg_rent
t = -8.2567, df = 69, p-value = 6.782e-12
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8057511 -0.5645144
sample estimates:
       cor
-0.7049711
```

Can you interpret the correlation coefficient - is it strongly/weakly positive/negative? Does the p-value indicate that the relationship between x and y is statistically significant?

A Pearson's correlation is only suitable when the relationship between the two variables is linear (shaped like a straight line). It is not sensitive to relationships that are non-linear, for example if y always increases with increasing x but not by a consistent amount). In these circumstances it is better to use the Spearman's rank correlation. This statistic is obtained by simply replacing the observations' values on the chosen variables with their ranked values within the sample before computing the correlation. This means it is also suitable for large scale ordinal variables.

```r
cor.test(Barc_neighbs$unemp, Barc_neighbs$avg_rent, method="spearman")
```

Do not worry if a red error warning about 'ties' appears - this message is cautionary and can be ignored. Does your conclusion about the relationship between these two variables change when using a Spearman's correlation compared with a Pearson's correlation, or are the two results quite similar?

Linear regression

Let's now fit a regression model where y is average rents in Barcelona's neighbourhoods. Simple linear regression uses a single straight line of predicted values as the model for the data. Another way of thinking about this line is as the best possible linear summary of the 2D cloud of points that are represented in the scatterplot you drew earlier. So if I were to tell you to draw a straight line that represents this pattern of points, then the regression line would be the one that best does it (if certain assumptions are met). This linear model thus takes the form of a straight line equation.

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

1. 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.

2. What is the slope of the line. The slope is just how much Y is predicted to change on average when X increases by 1 unit. The slope can be increasing (X has a positive association with Y) or decreasing (a negative association).

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

This equation contains the origin/intercept of the line (b0) and the slope (b1). To find these parameters or coefficients R *estimates* them from the data. For linear regression models, R uses a method called least squares estimation to estimate the coefficients that create the line with the minimum distance between the points in the scatterplot and the regression line. You should read up about this in Field et al (2012) pages 246-252.

In order to fit a linear regression model in R we use the **`lm`**`()` function using the formula `(Y ~ X)`. The ~ character simply means that we are modelling y as a function (~) of some independent x variables. Typically, after fitting your model you will want to store it in a new object so R can draw on it again later. For now let's call our model 'fit_1':

```
fit_1 <- lm(avg_rent ~ unemp, data=Barc_neighbs)
```

This model regresses avg_rent (our Y variable) on unemp (our X) using the Barc_neighbs dataset as the data object. After executing this code you should see a new object called fit_1 – this contains the regression output. Now let's use this fitted model to add the regression line to a scatter plot of our two variables with the **`abline`**() function. You will see that we are stringing two functions together here using a + symbol.

```
#Create a 2D scatter plot with a regression line

plot(Barc_neighbs$unemp, Barc_neighbs$avg_rent, xlab="% unemployed", ylab="Average
rent (€)") + abline(fit_1, col="red")
```

Does this line look appropriate? Are there any obvious outliers? Try adding a title to the plot, adjusting the point symbols and size and then saving it to your workspace using the same techniques you learnt earlier today.

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

```
summary(fit_1)
```

That will produce the output below:

```
Call:
lm(formula = avg_rent ~ unemp, data = Barc_neighbs)

Residuals:
    Min      1Q  Median      3Q     Max 
-311.12  -97.79  -36.80   75.08  759.97 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1148.139     60.253  19.055  < 2e-16 ***
unemp        -50.438      6.109  -8.257 6.78e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 159.6 on 69 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.497,	Adjusted R-squared:  0.4897 
F-statistic: 68.17 on 1 and 69 DF,  p-value: 6.782e-12
```

For now concentrate on the numbers in the "Estimate" column. The value of 1148.139 estimated for the intercept (b0) is the "predicted" value of Y when X equals zero. This is the predicted rent in € when the unemployment rate is 0%. As you can imagine, this information is not really very useful as in reality there are no neighbourhoods without anyone unemployed! It is possible to make the intercept more meaningful by 'centring' the x variable so that its 0 is a more sensible value but we don't need to do that today.

The estimate value for the 'unemp' coefficient is what we are primarily interested in. This value is -50.438. This coefficient has a convenient interpretation. It indicates the average change in y (rent) associated with a one-unit increase in x (the unemployment rate). Here, we can interpret the coefficient as "for every one percentage point increase in unemployed residents, on average monthly neighbourhood rents fall by just over €50." Think back to the Pearson's correlation coefficient you produced earlier – does this regression coefficient fit with this result?

Knowing these two parameters not only allows us to draw the regression line, we can also use the regression equation to *predict* y at any given value of X. If the unemployment rate is 5%, we can simply go back to our regression line equation and insert the estimated parameters to compute our predicted y at this point:

The regression equation: y = b0 + b1x

Our prediction for y when x=5: y=1148.139 + -50.438*5 = €896

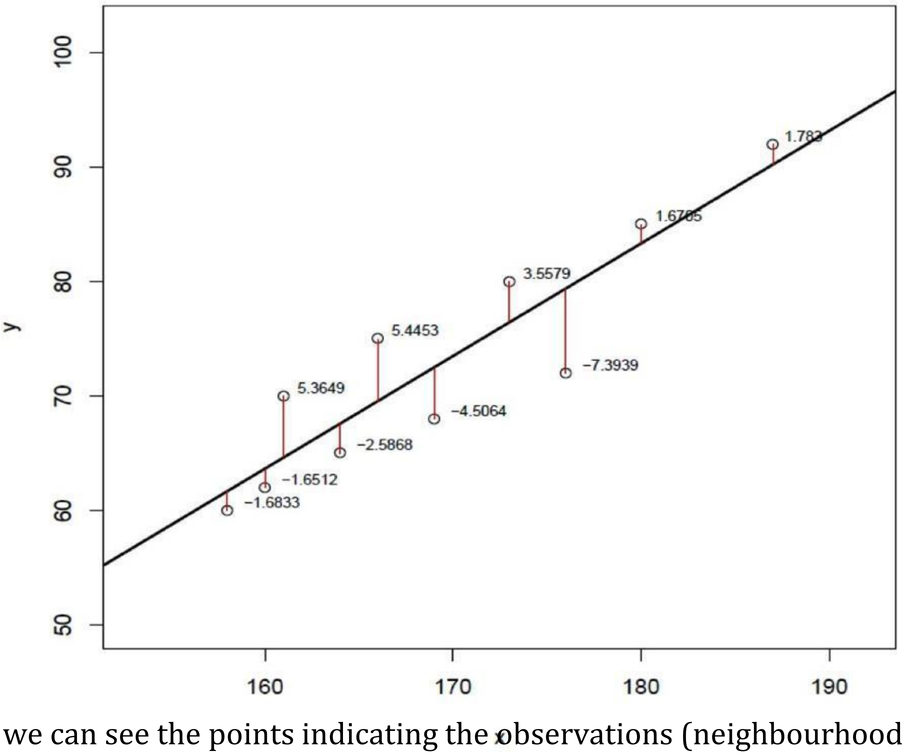
Or if you don't want to do the calculation yourself, you can use R's predict function:

```
predict(fit_1, data.frame(unemp = c(5))) #First you name your stored model and then you identify the new data (which has to be in a data frame format and with a variable name matching the one in the original data set)
```

On your own, use **predict()** or your computer's calculator to work out the predicted average rent for neighbourhoods with (i) 7% and (ii) 12% unemployed. Check your predictions make sense by looking back at the scatter plot with the regression line superimposed on it.

R squared and residuals

In the output above R reported values for the model ‘residuals’. The residuals are the differences between the observed value 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 value predicted for it by the regression line (see visual example below). Every time you tell R to fit a regression line it does this by finding the line that minimises the sum of these squared residuals.



In this graphic we can see the points indicating the observations (neighbourhoods) in our sample, with red lines indicating the distance between these points and the regression line (the red line is the residual for each point– quantified by the numbers next to the points). 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 by looking ONLY at the value of X. There are other things that matter for Y which are not being taken into account in our model. And then, of course, we have measurement error and other forms of ‘noise’ in the data.

The overall distribution of residuals captures how much variation in y is unexplained in our model. Larger residuals mean that our model is explaining less of the variation in y than if the residuals are small. This means that we can use information about the values and distribution of residuals to assess the performance of our model – how well it fits the data.

You should recall from your preparatory readings that the model r^2 tells us the proportion of the total variation in y that is being explained by our model. By reducing the residuals and improving the r^2 we aim to build a model that can explain more variation in y, in theory thereby producing a better representation of reality and thus better predictions. If you look at the R output you will see that the r^2 for our current model is 0.497(look at the multiple R_2 value on the bottom lines). Knowing how to interpret this is important. r^2 ranges from 0 to 1, with 0 representing no explanation and 1 perfect prediction. From our r^2 value we can say that our model explains about 49.7% of the total variation in the percentage of average rents across Barcelona.

Inference with regression

We often only have access to a subset of all possible observations from which to compute the least squares line (this subset is known as the ‘sample’). We want to use this sample data to make *inferences* about the parent population from which it is drawn, but we can’t know anything about the relationships between variables in this parent population because we don’t have enough data. So our regression line is only one of many that could be estimated. A different sample of neighbourhoods would probably produce a different regression line because of inherent variations in the data gathered from sample to sample. 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 sample estimates obtained over a very large number of data sets, the average of these estimates should 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 also compute standard errors for the regression coefficients to quantify our uncertainty about them. You can then use (i) this information about uncertainty as well as (ii) the size of the coefficient value to perform a hypothesis test of their statistical significance. For each coefficient, this procedure tests the null hypothesis that the true population value of the regression coefficient is actually 0 (i.e. that the x variable actually has no relationship with y in the population). The probability of finding a value of the coefficient in the sample at least as extreme as the one you found if this null hypothesis is true is then shown by the p-value. If the p-value is small, the logic is that you are extremely unlikely to have found a coefficient of this magnitude in your sample if the variable actually has no relationship to y in the population. To put it another way, a small p-value means that you would be extremely unlucky to find a coefficient of this magnitude in your sample if x in reality has no relationship with y. Bigger coefficients and smaller standard errors yield larger t-statistics and smaller p-values as we can be confident that large, precisely estimated coefficients are very unlikely to be found in sample data when the null hypothesis (that there is no relationship between X and Y) is actually true in the population.

In our example, we can see that the coefficient for our X variable has a very small p-value and so can be thought of as statistically significant (at both the 5% and 1% levels because it is smaller than both 0.05 and 0.01, the values marking these thresholds). Notice that the t statistics and p-value are the same as the correlation coefficient.

```
summary(fit_1)

cor.test(Barc_neighbs$unemp, Barc_neighbs$avg_rent)
```

Multiple regression

A key aim of many regression analyses is to build up a 'better' model by increasing the number of predictors (X variables). In our case we can also add the p_fcitizens variable into the model predicting average rents across Barcelona to see whether the % of residents with foreign citizenship has any link to the price of housing.

Another reason we typically want to include multiple independent variables is to ‘control’ for ‘confounding’ factors that might be associated with both X and Y and thus explain away their observed association. 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 between x and y. 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 it is these 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) that are correlated with both ethnicity and voting, and which in reality explain away observed ethnic patterns of centre-left voting?

We can test the relevance of competing explanations of variations in Y using multiple regression. If you find a significant coefficient for X that remains even after you control for potential confounders (selected using theory and prior research), then you can be more confident you have found a real ‘effect’ of X (note that even with regression finding a significant coefficient does not necessarily signify a causal pathway between X and Y!)

Let’s fit a multiple regression model using the **lm()** function. The code to do this is just a straightforward extension of the linear model we estimated before and stored as fit1.

```
fit_2 <- lm(avg_rent ~ unemp + p_fcitizen, data=Barc_neighbs)

summary(fit_2)
```

With more than one independent variable, you need to ask yourself whether the model you have fitted is actually a significant improvement over the null model (the model with no parameters where we just predict the mean of y for every observation). The null hypothesis that it is not an improvement (and thus that the coefficients are not different from 0) is tested with a F test. You see the F test printed at the bottom of the summary output. The F test statistic on its own isn’t very useful but it does come with an associated p-value. We can interpret this p-value in the usual way. In our model p is well below the conventional .05 that we use to declare statistical significance and reject the null hypothesis. At least one of our explanatory variables must be related to our dependent variable and thus not equal to 0.

Notice that the results table also reports a t-test and p-value for each predictor. In the multiple regression model, these are now testing whether each predictor is significantly associated with the response variable *after* adjusting for the other variables in the model. Look at these values for the unemp coefficient – you will see that they are not the same in the multiple regression model as they were in the simple linear model (which did not control for foreign citizenship rates). Equally the coefficient value is not the same. Indeed, looking at the model results shows that the p-value for the p_fcitizen variable is much greater than 0.05 and so only unemployment seems to be a statistically significant predictor of average rents.

If we look at the r^2 we can now see that it is also fractionally higher than before when the only independent variable was `unemp`. R2 will always increase as a consequence of adding new variables, even if the new variables are only weakly related to the response variable. The adjusted R2 avoids this problem because it controls for the number of variables we have included in our model. You should see that this r2 measure barely changes between the two models. Hence, it appears that the rate of foreign citizens in a neighbourhood has no significant links to average rents after controlling for unemployment levels. We might therefore want to think about dropping it to simplify our model.

If you have time, now try running scatterplots and a linear regression model without the unemp variable to see whether foreign citizenship rates have any bivariate links to rents in Barcelona.

Time to call it a day. You should be aware that regression models carry lots of assumptions that need to be satisfied for the results to be valid. You can read about them in the Field et al. (2012) reference.