Practical 3: Bivariate Plots 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 basic techniques used to create two-dimensional plots 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:

- Create a simple scatter plot
- Create a symbols plot
- Create plots in the ggplot package

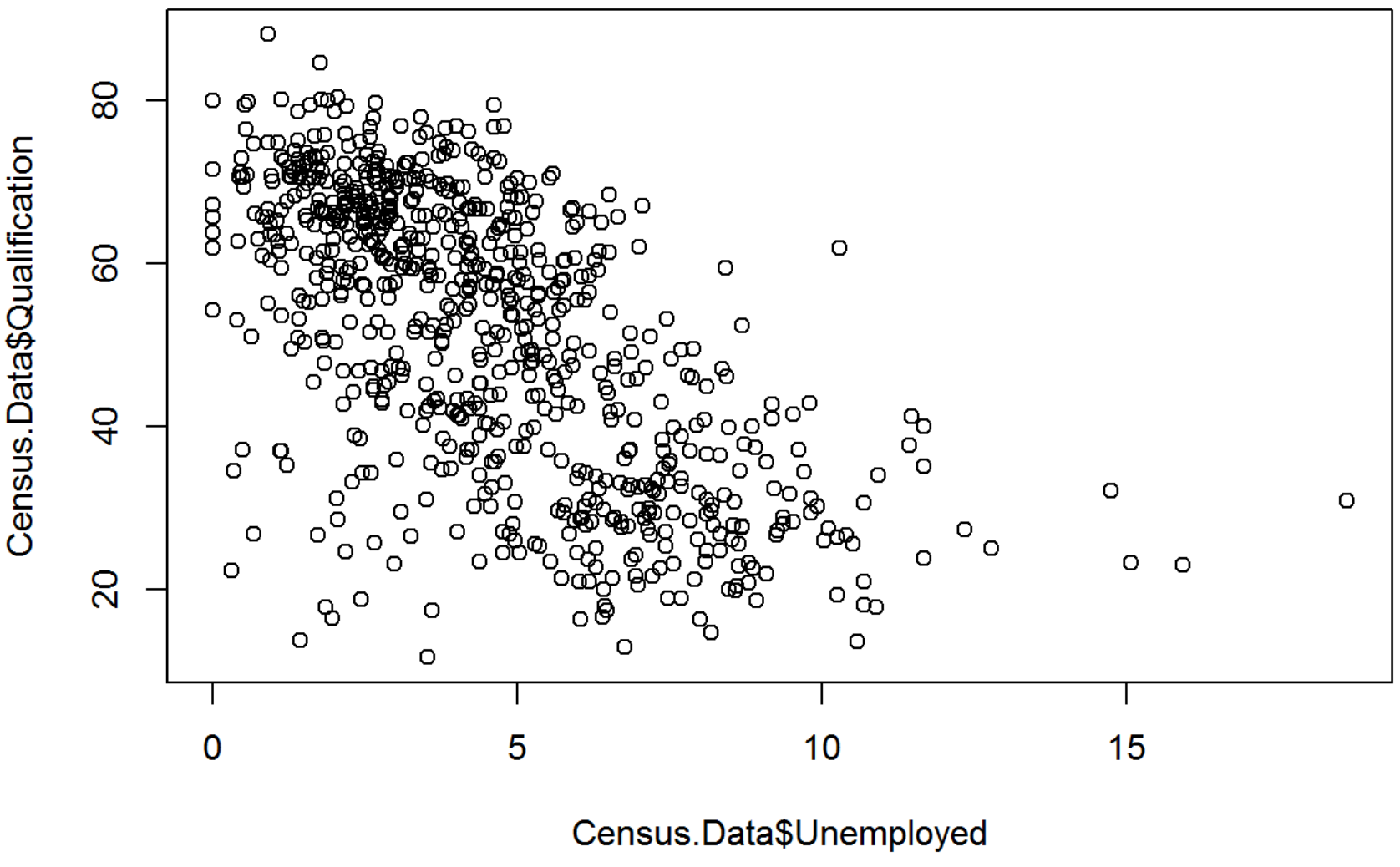
First, we must set the working directory and load the practical data.

```
#Set the working directory.
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")
```

Simple scatter plots

The most basic scatter plots (http://www.itl.nist.gov/div898/handbook/eda/section3/scatterp.htm) in R can be made using the `plot()` function which only requires you to identify two variables within the function's parameters. To do this follow the example below.

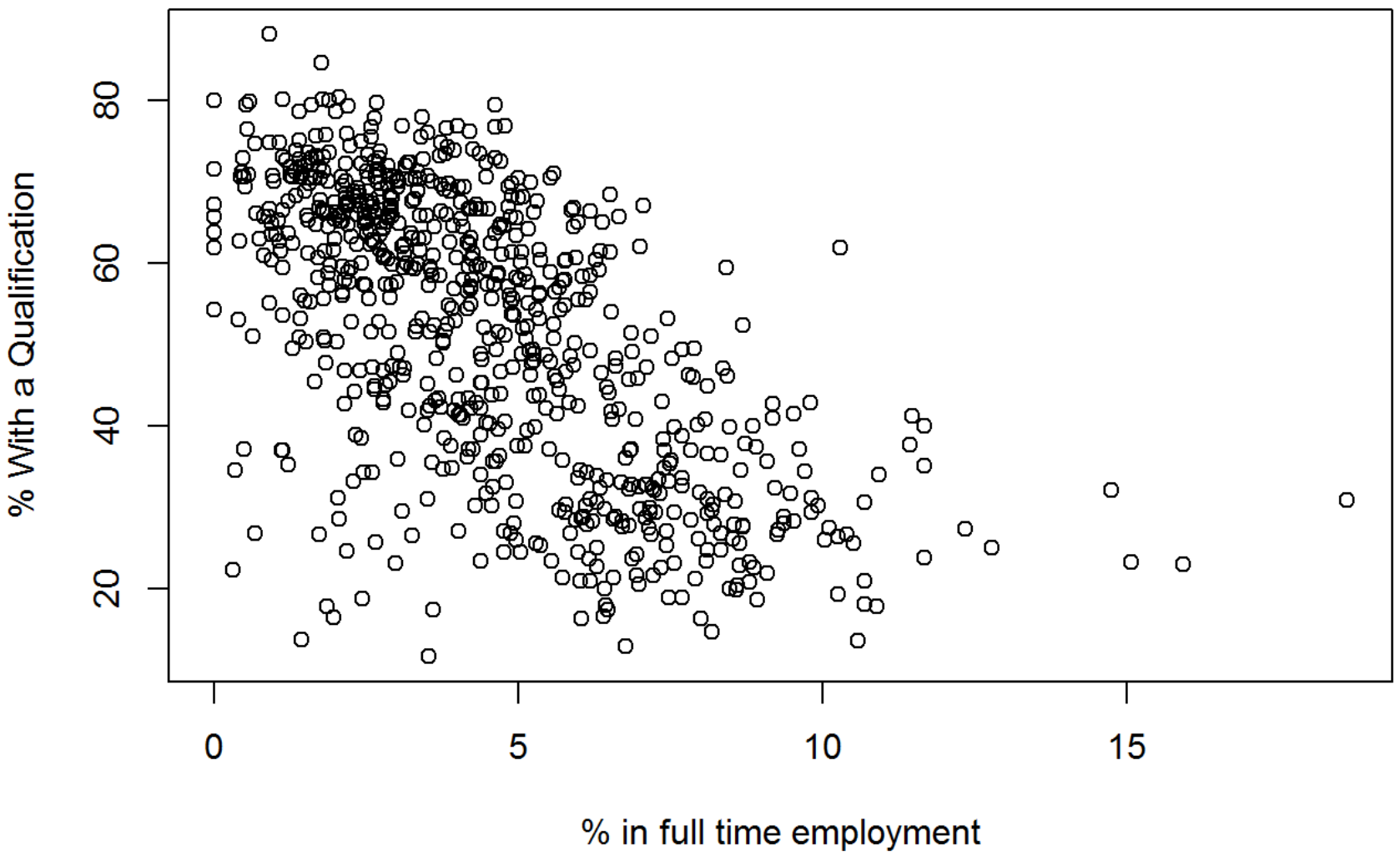
```
#left of the comma is the x-axis, right is the y-axis. We are using the $ command
to select the columns of the data frame we want

plot(Census.Data$Unemployed,Census.Data$Qualification)
```



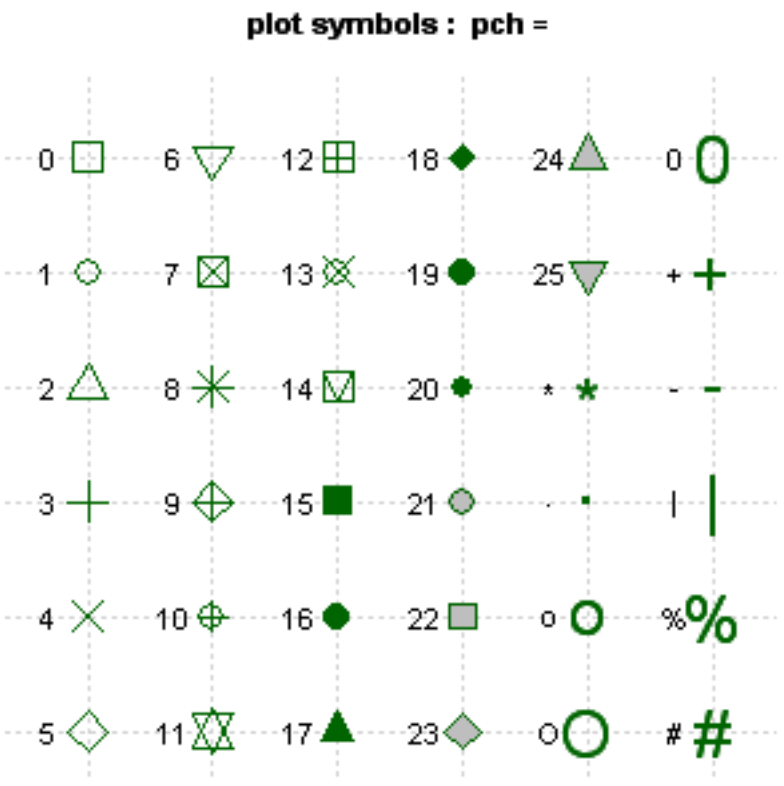
Each point represents data from the two variables for a unique output area. From this chart, it is possible to infer that there is a distinctive negative relationship between the percentage of those with *level 4 qualifications and above* and the percentage of those *unemployed* at the output area level. As level 4 qualifications refer to certificates of higher education we could expect there to be an inverse relationship between this variable and local unemployment rates.

Remember to observe the available parameters for a function we can enter `?plot` into R. We will now include axis labels.

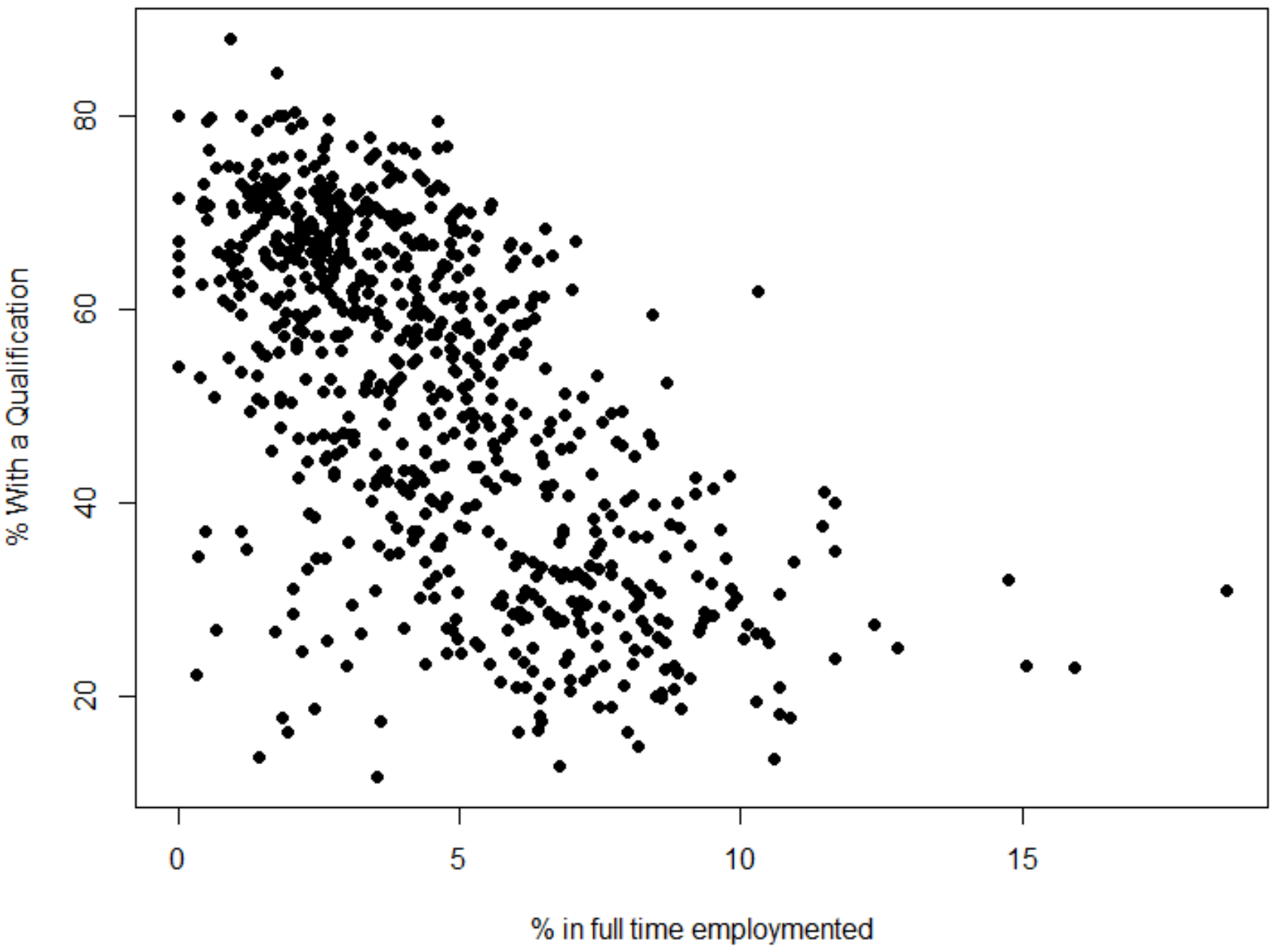
```
# includes axis labels
plot(Census.Data$Unemployed,Census.Data$Qualification,xlab="% in full time employm
ent", ylab="% With a Qualification")
```



One of the parameters for the `plot()` function is plotting symbols (pch). Details of what symbols are available can be found on the following webpage: http://www.statmethods.net/advgraphs/parameters.html (http://www.statmethods.net/advgraphs/parameters.html)



If you want to change the default of hollow circles to filled circles, you can just add `pch = 19` to the end of the function (remember to insert a comma after the previous command). The output of this is demonstrated below. Try changing the plots to triangles with your data.



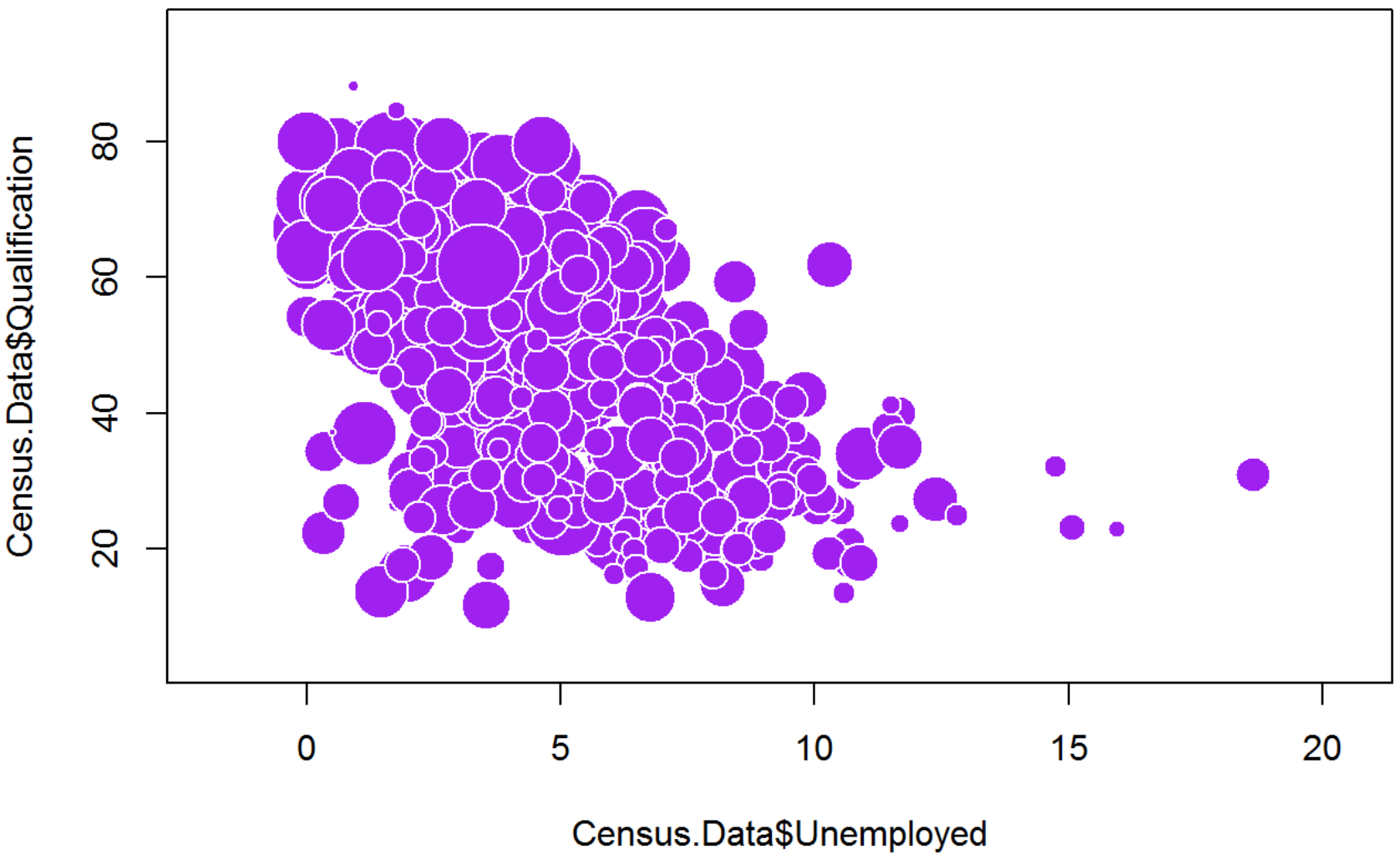
Symbols plot

There is also the possibility of creating a proportional symbols plot using the `symbols()` function which shares much of the parameters as the `plots()` function. This allows us to consider a third dimension (aka a third variable) in our two-dimensional plot.

In this example, we will set the *percentage of White British persons* as the size dependent variable.

Now using the `symbols()` function we will set the symbols as circles and use the *percentage of White British persons* to define the proportional sizes of them in the chart. We have also defined the foreground(`fg`) and background(`bg`) colours of the symbols. The inches command allows you to restrict the overall size of all the symbols. Try changing some of these commands below.

```
# Create a proportional symbols plot
symbols(Census.Data$Unemployed,Census.Data$Qualification,  circles = Census.Data$W
hite_British, fg="white", bg ="purple", inches = 0.2)
```



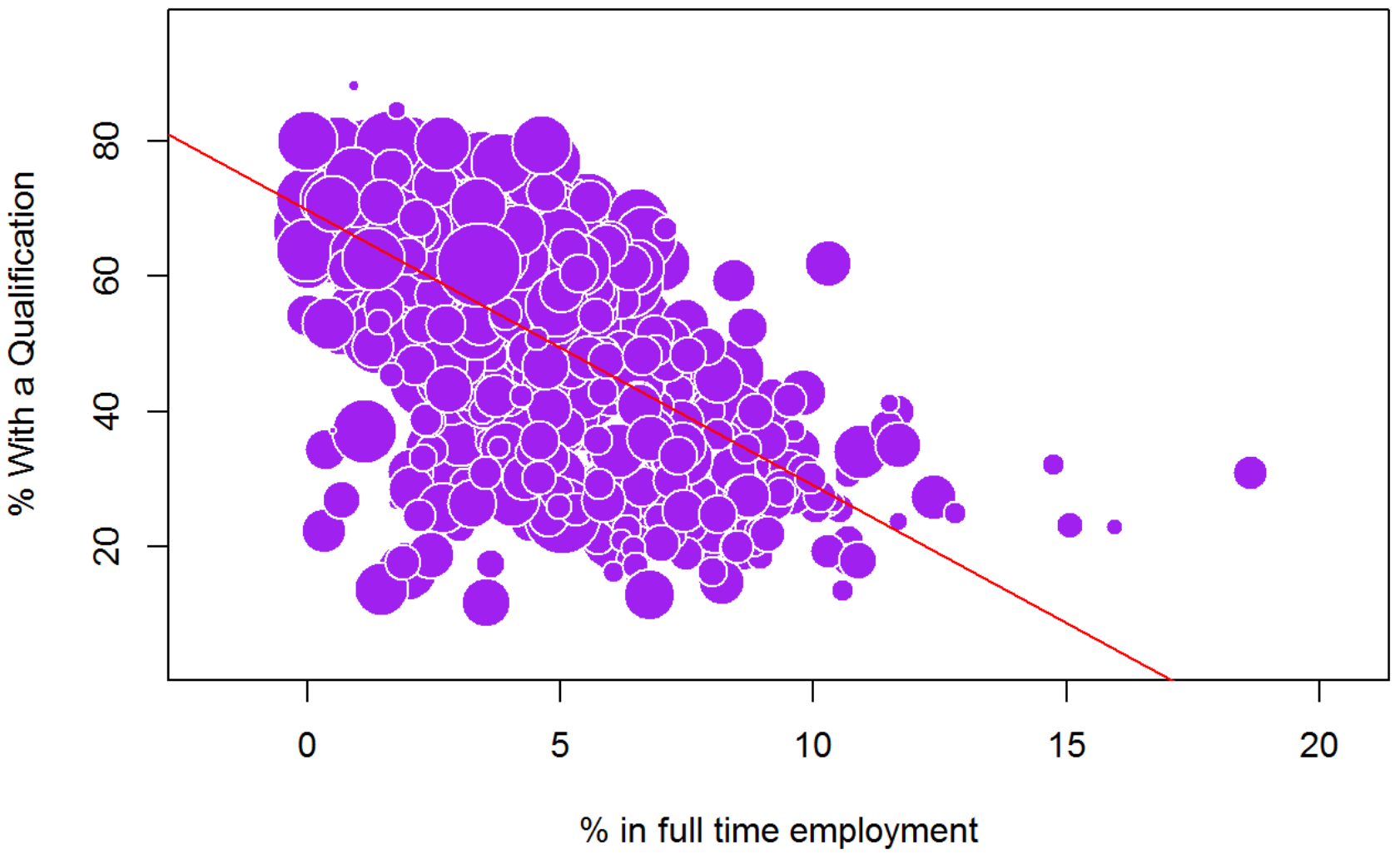
In the graphic, larger circles represent areas of greater percentages of the White British population. Can you depict a relationship between this variable and the other two in the plot?

There are still improvements that can be made to the plot. For instance adding x-axis and y-axis labels (`xlab` , `ylab`).

Adding a regression line

It is also possible to insert a regression line (http://www.statsref.com/HTML/index.html?regression.html) to plots in R. This requires you to run a quick linear regression model which you can use to plot a straight line of best fit in the chart. This can be done in R by adding (+) a linear model (`lm()`) function to our plot. The outputted regression line is then represented using the 'abline()' function. We will cover the regression model in more detail in a later practical.

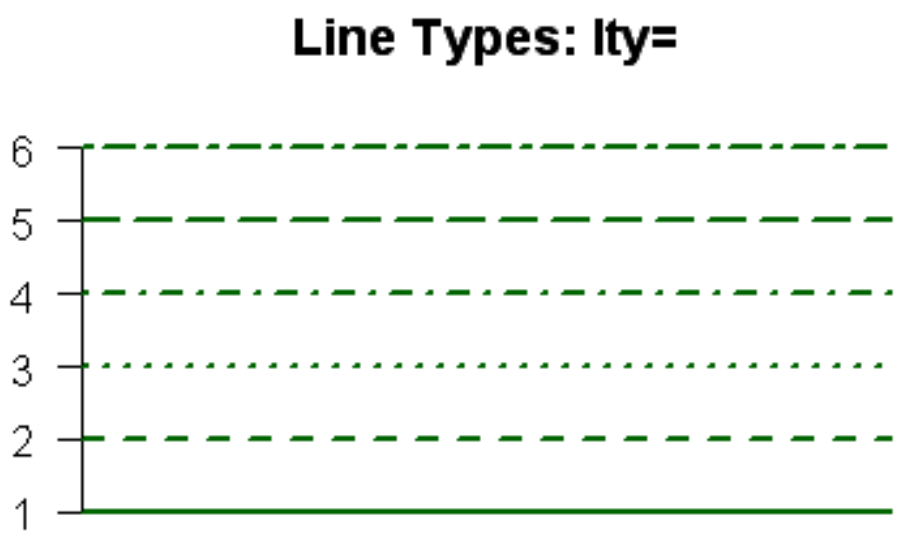
```
# bubble plot
symbols(Census.Data$Unemployed, Census.Data$Qualification,  circles = Census.Data$
White_British, fg="white", bg ="purple", inches = 0.2,  xlab="% in full time emplo
yment", ylab="% With a Qualification") +
# adds a regression line, sets the colour to red
abline(lm(Census.Data$Qualification~ Census.Data$Unemployed), col="red")
```



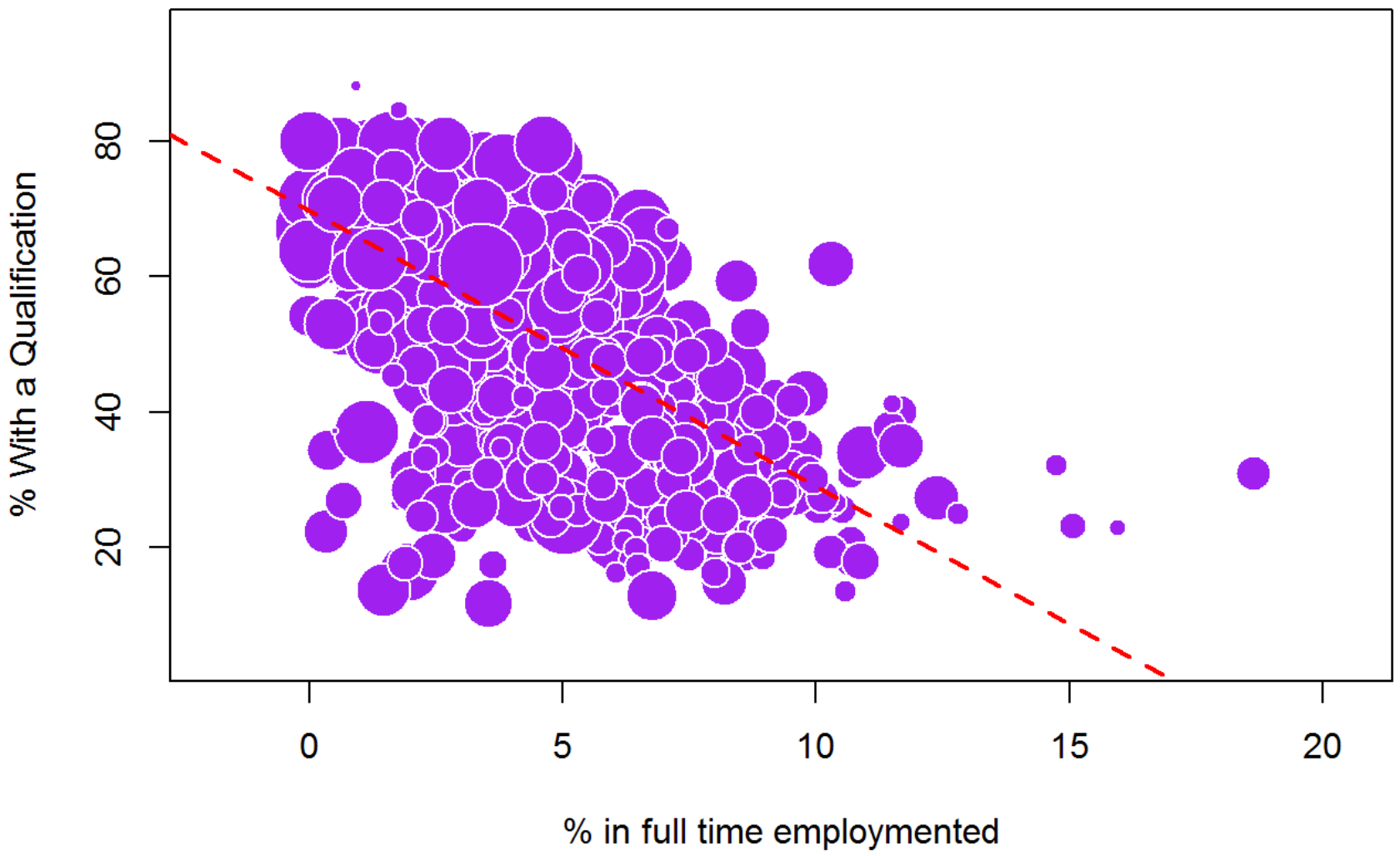

```
## numeric(0)
```

We can also edit the line using the line type (`lty`) and line width (`lwd`) commands from the `abline()` function. These are demonstrated below using an image from http://www.statmethods.net/advgraphs/parameters.html (http://www.statmethods.net/advgraphs/parameters.html)

option	description
lty	line type. see the chart below.
lwd	line width relative to the default (default=1). 2 is twice as wide.



```
# a bubble plot with a dotted regression line
symbols(Census.Data$Unemployed, Census.Data$Qualification,  circles = Census.Data$
White_British, fg="white", bg ="purple", inches = 0.2,  xlab="% in full time emplo
ymented", ylab="% With a Qualification") + abline(lm(Census.Data$Qualification~ Ce
nsus.Data$Unemployed), col="red", lwd=2, lty=2)
```




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

Using the ggplot2 package

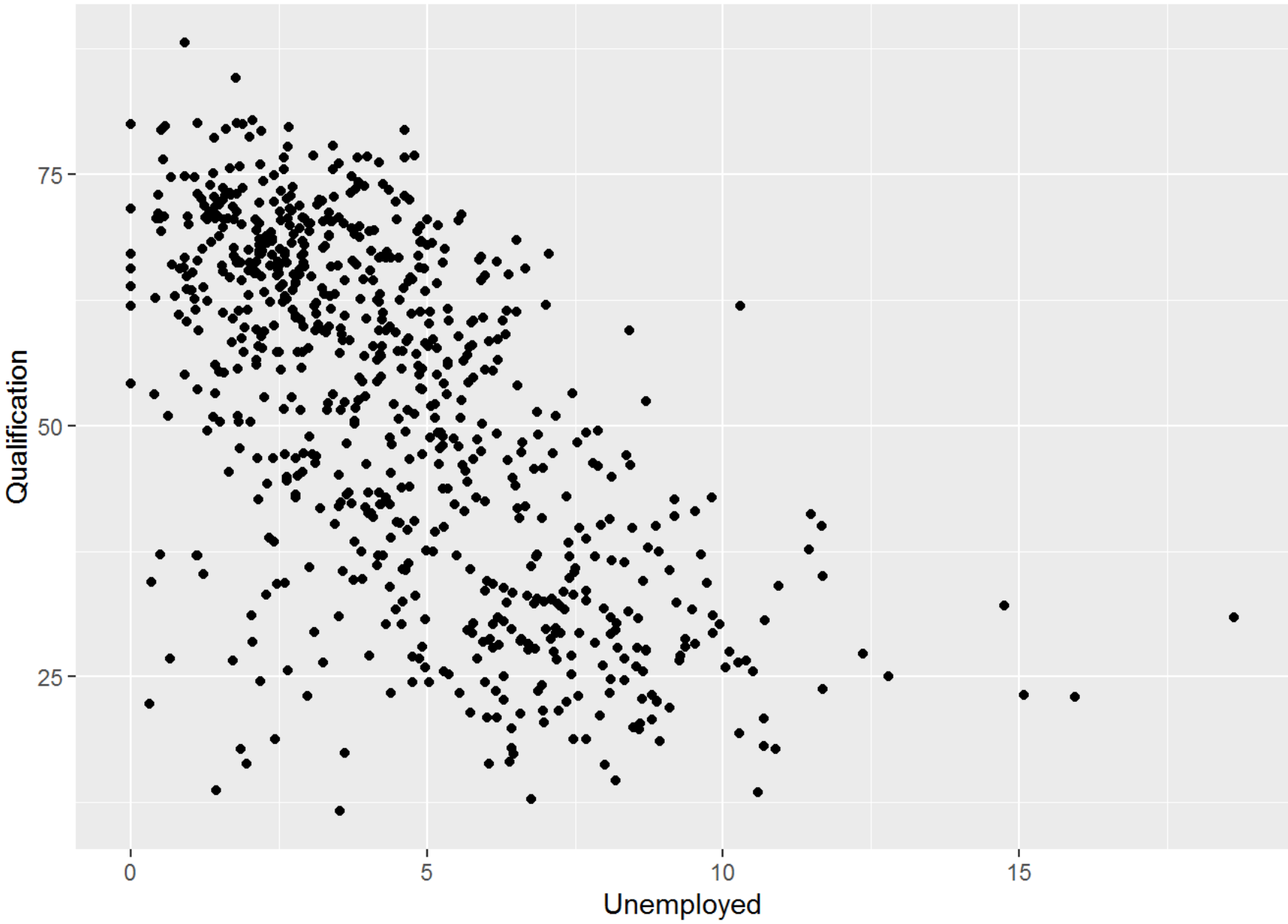
A slightly different method of creating plots in R requires the ggplot2 package. As the commands from this package are not already in R, we need to install ggplot2 and then load it. Look at your notes from Practical 2 about how to install packages.

```
# Loads an installed package
library("ggplot2")
```

The package is an implementation of the *Grammar of Graphics* (Wilkinson, 2005) - a general scheme for data visualisation that breaks up graphs into semantic components such as scales and layers. ggplot2 can serve as a replacement for the base graphics in R and contains a number of default options that match good visualisation practice.

We will first create a simple scatter plot using `ggplot()`

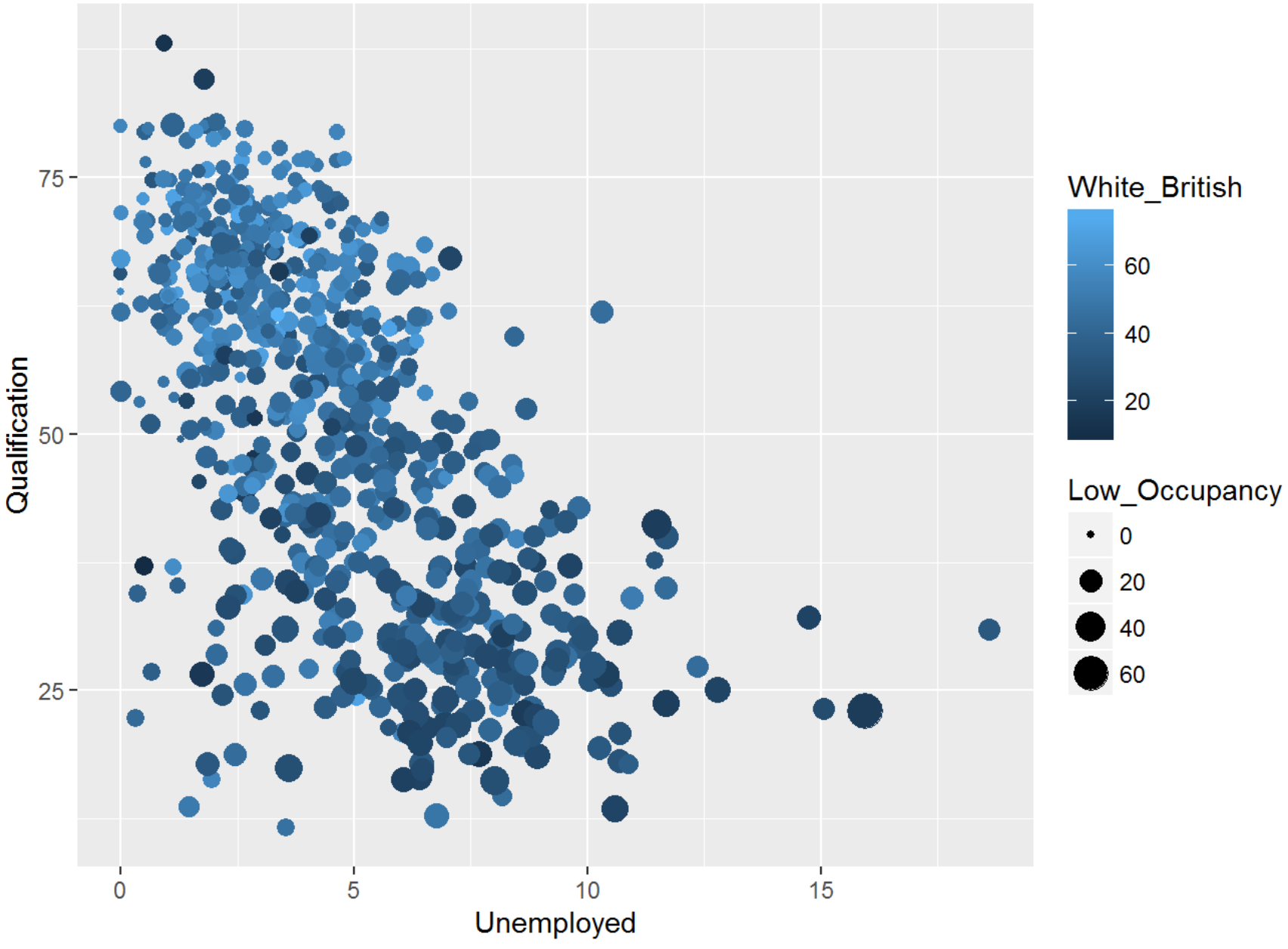
```
p <- ggplot(Census.Data, aes(Unemployed,Qualification))
p + geom_point()
```



Notice that here we assign the data object, then in the `aes()` command we set the two variables. The points are also inputted as a separate function to the `ggplot()` function.

We can also set various parameters for the points such size and colour. For example in the code below the colours are proportional to the percentage of the White British population, whilst the size is proportional to the percentage of homes with a low occupation rating (i.e. overcrowded). Here it is possible to observe four different variables in one two-dimensional chart.

```
p <- ggplot(Census.Data, aes(Unemployed,Qualification))
p + geom_point(aes(colour = White_British, size = Low_Occupancy))
```



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)