Worksheet 3

Linear regression in R

Timetable week: 6  
Topic: "Linear models"

# Learning outcomes

By the end of the session you will know how to:

* Fit and summarise linear regression models in R
* Interpret results from linear regression models
* Understand the main assumptions behind linear regression models

# Intro

One of the most commonly encountered statistical methods in social science publications is **linear regression**. This week we will make some first steps towards understanding what this method involves and applying it in practice using the R programming language.

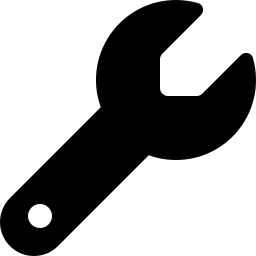
We begin with the simple (*but not trivial!* - as (Gelman, Hill, and Vehtari 2020) admonish - case where we model the relationship between two ***numeric*** (***continuous***) variables using a ***simple linear regression model***. The aim of a **simple linear regression model** is to predict the expected values of one ***numeric*** variable (e.g. a child’s *age*) from another ***numeric*** variable (e.g. the child’s *height*). The variable we want to predict is usually called the *response* (or *outcome*/*dependent*/*explained* variable) and the variable used for predicting this outcome is referred to as the *explanatory* (or *predictor*/*independent* variable).

In actual applied sociological research the *simple linear regression* model is rarely used on its own because we understand that in the social world there are always many factors that influence an outcome at the same time - even in the case of modelling a child’s *height*, we know that *age* is an important factor, but there is some considerable variation in height even among children of the same age, so there must be other factors at play too. We also understand that assuming a strict ***linear*** relationship between two variables is too much of an oversimplification (staying with the *height ~ age* example, while the linear assumption may be realistic when applied to children, it is definitely not applicable if we extend the analysis to all ages: an increase in age is associated with an increase in height for children, but once the age of maturity is reached, we no longer get taller as we get older - in fact, in old age, as we get older we tend to get shorter!).

But the simple linear regression model contains many of the statistical components on which other statistical tests and more complex models are built, so understanding it is essential. We will also learn a few concepts and methods related to simple linear regression - such as *correlation* (also known as the *Pearson correlation,* after the mathematician [Karl Pearson](https://en.wikipedia.org/wiki/Karl_Pearson)) - and we’ll make some steps towards expanding the linear model to include non-continuous predictors. The main R function that we will learn is lm() (check help(lm) for technical information).

# Exercise 0: Setup

1. **Open the R Studio interface** by clicking on the ***.Rproj*** file included in the project folder that you created in *Lab2*. The folder should be stored on your Newcastle University **OneDrive** and accessible from any computer.
2. **Create a new blank Quarto document for this lab session and call it *Lab3.qmd***
3. At the top of the page briefly detail what the script is about (e.g. “Lab work for Week x”).
4. **Load R packages** that we commonly use with the library() function

*  **Tip:** You may need to first install the package with the install.packages() function if it’s not yet installed
* library(tidyverse)  
  library(easystats)  
  library(gtsummary)  
  library(ggformula)

# Exercise 1: Is *inequality* associated with *generalised trust*?

About 30 minutes

For all data analysis tasks, follow a “0 + 5”-step workflow. First (**“Step 0”**, because it’s not really a separate analytical step), state the research question you are attempting to answer and identify data that can be used to address the question; then (**Step 1**) describe the variables (data) that you plan to use to answer your question; then (**Step 2**), if needed, modify (wrangle) any variables that need to be adjusted to help the analysis and interpretation; then (**Step 3**) describe how your *response* variable(s) and main *explanatory* variable(s) are related. This will help with the interpretation of the statistical results and with identifying any further changes to the variables that may be needed (returning to **Step 2** again). You may also wish to check the relationship between your predictor variables if you have more than one (but we’re not using multiple predictors yet); then (**Step 4**) apply the statistical model that is most appropriate to answer the research question; and finally (**Step 5**) summarise the results from your analysis using tables, figures and your own words.

## Research question and data

As a first exercise, let’s reproduce the analysis presented in the lecture. The exercise explores the relationship between inequality and generalised trust at the national level in international comparison. The aim is to assess the arguments put forward by (Wilkinson and Pickett 2010) in their Chapter 4. Their measurement of *generalised trust* comes from the World Values Study (1999-2001), while for the measure of inequality they use the so-called “S80/S20” measure, which is the ratio of the average income of the 20% richest to the 20% poorest people in a country. We have similar data obtained from the latest waves of the World Values Survey and European Values Study (2017-2022) and the [World Bank](https://data.worldbank.org/indicator/SI.DST.05TH.20). To ensure that inequality data is available for as many countries as possible, I calculated the “S80/S20” measure as the average of the data available in each country between the years 2010-2022.

The dataset can be loaded into R directly from the module’s github repository:

inequality <- data\_read("https://github.com/CGMoreh/SOC2069/raw/main/Data/for\_analysis/lab3macro.rds")

We can check the list of variables in the dataset and their first few values with the data\_peek() function from {datawizard}:

data\_peek(inequality)

Data frame with 89 rows and 10 variables  
  
Variable | Type | Values   
-----------------------------------------------------------------------  
cntry\_AN | character | AL, AZ, AT, AM, BA, BG, BY, HR, CZ, DK, ...  
country | character | Albania, Azerbaijan, Austria, Armenia, ...   
trust\_pct | numeric | 2.52, 30.36, 48.46, 17.68, 9.61, 18.05, ...  
GDPpercap2 | numeric | 12770.9918634405, 14121.4069355591, ...   
pop | numeric | 2873457, 9854033, 8797566, 2851923, ...   
urban\_pop\_pct | numeric | 59.383, 55.343, 58.094, 63.103, 47.876, ...  
inc\_top20 | numeric | 39.6375, NA, 38.4181818181818, 39.575, ...   
inc\_bottom20 | numeric | 7.9125, NA, 7.90909090909091, ...   
s80s20 | numeric | 5.00947867298578, NA, 4.85747126436782, ...  
Region | character | Europe & Central Asia, ...

## Describe individual variables

For summary statistics of the numeric variables in the dataset, we can use describe\_distribution():

describe\_distribution(inequality)

Variable | Mean | SD | IQR | Range | Skewness | Kurtosis | n | n\_Missing  
------------------------------------------------------------------------------------------------------------  
trust\_pct | 25.73 | 18.41 | 21.14 | [2.14, 77.42] | 1.13 | 0.62 | 89 | 0  
GDPpercap2 | 27447.01 | 21342.56 | 28284.55 | [1987.97, 1.23e+05] | 1.62 | 4.20 | 86 | 3  
pop | 5.78e+07 | 1.58e+08 | 4.68e+07 | [73837.00, 1.40e+09] | 7.32 | 61.13 | 87 | 2  
urban\_pop\_pct | 67.92 | 18.93 | 25.93 | [20.31, 100.00] | -0.49 | -0.38 | 87 | 2  
inc\_top20 | 42.69 | 5.09 | 6.80 | [34.56, 57.35] | 0.86 | 0.42 | 79 | 10  
inc\_bottom20 | 7.15 | 1.62 | 2.32 | [3.45, 10.08] | -0.19 | -0.69 | 79 | 10  
s80s20 | 6.50 | 2.60 | 2.79 | [3.52, 16.64] | 1.67 | 3.50 | 79 | 10

**Questions**

Examine the descriptive results and try to answer these questions:

* What is the average (mean) generalised trust score of the countries in the dataset?
* What is the average GDP per capita of the countries in the dataset?
* What is the average value for the measure of inequality across all the countries in the data?
* How spread out are the inequality scores? (**tip**: the *standard deviation* (SD) is a good measure of “spread”, or *variation* around the *mean*)
* What is the *minimum* and *maximum* percentage of the urban (versus rural) population in individual countries in the data? Do you think this value is correct, or might there be something wrong with the data? How would you explain this distribution? (**tip**: check the dataset itself to explore the individual data points; you can open it in a viewer window by double-clicking on the data object in the Environment, or using the View() function)
* Are there any missing values (NAs) on any of the variables in the dataset?
* How would you describe the distribution of the *population* variable?

As we have practised in Lab2, we can also tabulate factors and character variables with the data\_tabulate() function. For example:

inequality |> data\_tabulate(Region)

Region (Region) <character>  
# total N=89 valid N=86  
  
Value | N | Raw % | Valid % | Cumulative %  
---------------------------+----+-------+---------+-------------  
East Asia & Pacific | 15 | 16.85 | 17.44 | 17.44  
Europe & Central Asia | 42 | 47.19 | 48.84 | 66.28  
Latin America & Caribbean | 12 | 13.48 | 13.95 | 80.23  
Middle East & North Africa | 8 | 8.99 | 9.30 | 89.53  
North America | 2 | 2.25 | 2.33 | 91.86  
South Asia | 3 | 3.37 | 3.49 | 95.35  
Sub-Saharan Africa | 4 | 4.49 | 4.65 | 100.00  
<NA> | 3 | 3.37 | <NA> | <NA>

We see that the majority of the countries in the dataset are from Europe and Central Asia.

More importantly, let’s explore visually the distribution of our two main variable of interest: *generalised trust* and *inequality*.

Using the plotting functions we learnt in Lab2, create a histogram or a density plot for each of the two variables trust\_pct and s80s20:

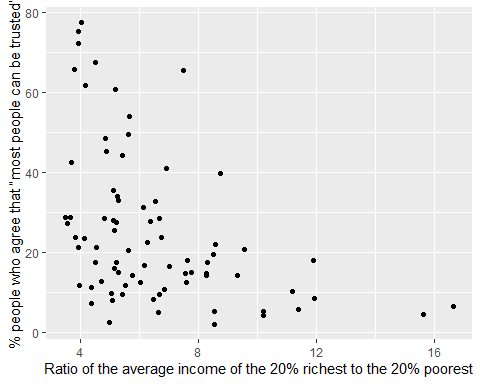
# Insert a new code chunk into your Quarto document, write your functions there and execute the code chunk to see the results

**Question**: would you describe the two variables as “normally distributed”?

## Describe the relationship between your variables

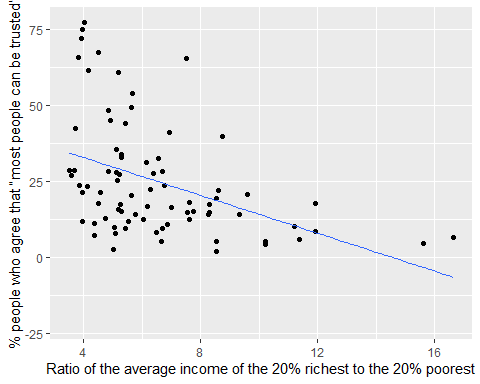
To visualise the relationship between two numeric variables we can use a *scatterplot*. Relying on the {ggformula} package we got to know in Lab2, we can use the gf\_point() function. We keep in mind that our *response* variable is *generalised trust* and the *explanatory* variable is *inequality*. So it is customary to place the *response* on the *y* axis and the *explanatory* (predictor) variable on the *x* axis, in line with the general formula notation ***y*** **~ *x***. The command is the following:

gf\_point(trust\_pct ~ s80s20, data = inequality)



To add a linear regression line to the scatterplot we can use the gf\_lm() function:

gf\_point(trust\_pct ~ s80s20, data = inequality) |>   
 gf\_lm()

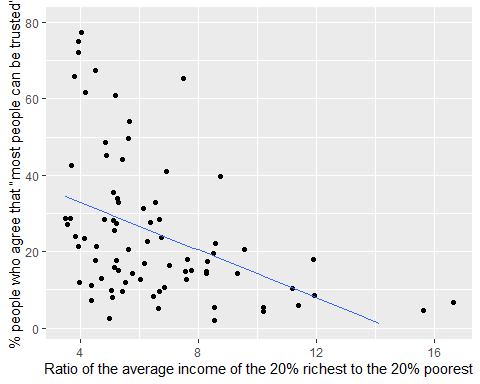


The straight line we plotted is a *regression line* that visualises the linear model (see **Agresti 2018: 250-252**; **Gelman et al. 2020: 37-38, 82-95**). In Step 4 we will fit this model statistically to obtain the values for the and coefficients so we can interpret the regression line more accurately. For the moment, let’s rely on our eyes.

One thing to notice immediately is how the y-axis has expanded into negative values. This is because the linear assumption of the regression line means that at high values of *inequality* very low, below-zero values of *trust* are predicted, even though that is not logically possible, given that our measurement is on a percentage scale (i.e. running between 0 and 100).

To constrain the *y* axis to the actual values of our data, we can set limits on the scale using the gf\_lims() function:

gf\_point(trust\_pct ~ s80s20, data = inequality) |>   
 gf\_lims(y = c(1, 80)) |>   
 gf\_lm()

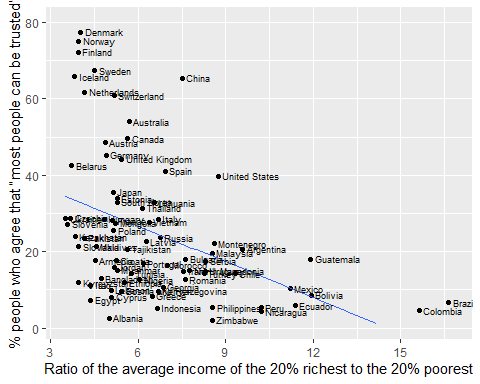


**Questions**

* Does the graph show a *negative* or *positive* correlation?
* Does the graph show a *strong* or *weak* correlation?

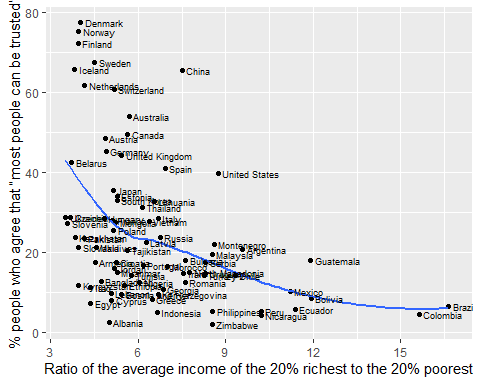
Another addition to the chart that may come useful are value labels, in this case the names of the countries; this would allow us to identify which are the outliers. We can use the gf\_text() function for this, with some additional specifications to set the size of the labels and their distance from the value points:

gf\_point(trust\_pct ~ s80s20, data = inequality) |>   
 gf\_lm() |>   
 gf\_lims(y = c(1, 80)) |>   
 gf\_text(label = ~ country, size = 2.5, hjust = 0, nudge\_x = 0.15, nudge\_y = 0.15)



We may wonder whether a linear function is really appropriate for our data, and we could replace the linear model line with a smoothed line that follows the shape of the data more closely:

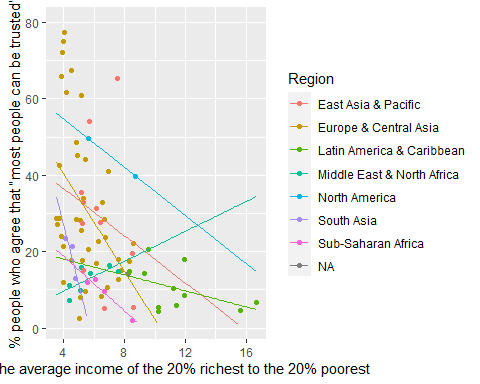
gf\_point(trust\_pct ~ s80s20, data = inequality) |>   
 gf\_smooth() |>   
 gf\_text(label = ~ country, size = 2.5, hjust = 0, nudge\_x = 0.15)



In this case, while it is apparent that the relationship between *inequality* and *trust* is not completely linear, the curvature is not too extreme.

Finally, we may wonder whether there are any group effects in the data that are being missed in the bivariate plot. We could add a third variable to the plot by using colouring. Below, we use the Region variable a third grouping variable (notice the ~ sign when adding the colour aesthetic):

gf\_point(trust\_pct ~ s80s20, data = inequality, color = ~ Region) |>   
 gf\_lims(y = c(1, 80)) |>   
 gf\_lm()



By adding a third variable to our visual model, the linear regression line also breaks down by Region and we can see more clearly that the effect (and indeed the direction) of inequality on trust is not the same within each regional grouping; in some regions the association is very steep and negative, while in others is flatter - or, in the case of “Middle East and North Africa” it is strong positive association.

This hints to an important limitation of correlations and simple bivariate regressions. It is often the case that third (and further) variables can have a very strong influence on the associations we observe, and we need to pay careful attention to disentangling the meaning behind our regression results.

## Fit the statistical model

So far we have only visualised the relationship between *inequality* and *trust*, but we haven’t build a statistical model to obtain the coefficients describing the relationship. We already know from the scatterplots that a linear model may not be the ideal approach for our variables, but it is a very simple model that is worth starting with. In R, we can use the lm() function to run a **l**inear **m**odel. Because we often want to have access to various components of the models we fit for further analysis, it’s recommended to always save the model results to an object. Let’s call out first model **m1.1**:

m1.1 <- lm(trust\_pct ~ s80s20, data = inequality)

**Coding tip: Understand and simplify the formula**

Click to view

The lm() formula echoes the mathematical equation we are fitting:

We have data on the blue components (our two variables) and we are searching for the corresponding coefficients for the red components. The is called the “**intercept**”, referring to the point where the regression fit line we plotted earlier on the scatterplot *intercepts* the (trust) axis **when the** (inequality) axis equals **0**. The is called the *slope* (visually, it’s the angle between a completely flat horizontal line and the regression fit line) and it tells us the observed difference between different countries’ values of when their levels of also differ by one unit.

The *intercept* is included in the lm() function by default, so we don’t need to explicitly write it. We also don’t have to explicitly write data =, we can simply write just the name of the dataset. Our function call could be simplified to:

m1.1 <- lm(trust\_pct ~ s80s20, inequality)

However, leaving out data = does not work for all formula-style functions, so until you become very used to writing these commands, it’s better to spell out more than the strictly necessary. Also, remember that it’s more important to be accurate and to make it easy for yourself and others to understand your code easily.

We can see that the object “m1.1” now appears in the Environment pane, and if we expand it we see a list of different elements (“coefficients”, “residuals”, “effects”, etc.). The “coefficients” are the most elementary and most important elements. We can print them by using the coefficients() function or simply coef():

coef(m1.1)

(Intercept) s80s20   
 45.384383 -3.114425

As noted in the *Coding tip* above, the two *coefficients* that we see here for (Intercept) and s80s20 are the corresponding values for and , respectively, in the equation. We can substitute them - rounding down to two decimal places for simplicity - to obtain the equation: . So what does this mean?

We said at the start that the aim of a linear regression model is to help us predict the value of *generalised trust* from that of *inequality*; i.e. what would our best guess concerning a country’s population’s level of *trust* be if the only information we had about that country was its level of inequality? Knowing a country’s *inequality* score is already more information than ***not*** knowing anything. Without information on *inequality* our best guess would be simply the overall ***mean*** (i.e. ***average***) trust score for the whole dataset. If we remember from the descriptive statistics we’ve done earlier in the exercise, the **mean** of the trust variable was 25.73.

We can actually obtain the mean of a numeric variable also by using the lm() function without any predictors specified. Let’s check the mean *trust* score using this method and compare it with what we got from the summary statistics earlier:

lm(trust\_pct ~ 1, data = inequality) |> coef()

(Intercept)   
 25.73056

The coefficient is exactly the same as the one we got from the descriptive statistics we did in earlier steps. So, without knowledge of a country’s *inequality* level, or any other further information, our safest guess of a national population’s *trust* level would be 25.73. However, in knowledge of a country’s *inequality*, the slope () coefficients we got from the regression tell us that the best guess is the *intercept* (45.38) **plus** -3.11 **times** the country’s inequality score. For example, our best guess for the *generalised trust* score of a country with an inequality score of *5.4* in our dataset would be:

45.38 + (-3.11 \* 5.4)

[1] 28.586

More generally, the model tells us that a 1-unit difference in *inequality* is associated with a 3.11-unit difference in *generalised trust* as measured here.

Remember how we said earlier that the *Intercept* is the point where the regression fit line intercepts the axis when the axis is 0? This may sound okay visually, but note that in the case of our variables this means that the *Intercept* refers to the value of *trust* when the *inequality* variables is equal to **0**. Mathematically, there’s nothing wrong with this, but in practice we don’t have countries scoring “0” on *inequality* in our dataset - in fact, a S80/S20 income quintile share ratio of “0” would not make much sense at all. In other words: the interpretation of the *Intercept* value is meaningless in our case. That doesn’t affect the *slope* coefficient, so our regression equation is useful for predicting from, but we should not interpret the *Intercept* as it is.

In order to make the *Intercept* more meaningful, one common approach is to **mean-centre** the predictor variable(s). In our case, we would “shift” the *inequality* scale so that the average (i.e. mean) inequality takes the value of “0”. The mean s80s20 inequality score in the dataset is 6.5, so using the mean-centred variable in the regression would make the *Intercept* refer to the value of *inequality* for the countries with an average income quintile share ratio; of course, there may not exist such a country in the dataset, but the value itself is theoretically possible within the context of our measurement.

To request a more detailed summary of the model results, the base R function to use is summary():

summary(m1.1)

Call:  
lm(formula = trust\_pct ~ s80s20, data = inequality)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-27.263 -12.432 -3.831 7.003 44.597   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 45.3844 5.2088 8.713 4.29e-13 \*\*\*  
s80s20 -3.1144 0.7444 -4.184 7.53e-05 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 17.07 on 77 degrees of freedom  
 (10 observations deleted due to missingness)  
Multiple R-squared: 0.1852, Adjusted R-squared: 0.1747   
F-statistic: 17.51 on 1 and 77 DF, p-value: 7.527e-05

The summary() function prints out a lot of information, but it’s not the best format if we wish to reuse the various statistical components for further analysis, and the presentation of the output could also be improved. The model\_parameters() and the model\_performance() functions from the {parameters} package part of {easystats} is a better option:

model\_parameters(m1.1)

Parameter | Coefficient | SE | 95% CI | t(77) | p  
------------------------------------------------------------------  
(Intercept) | 45.38 | 5.21 | [35.01, 55.76] | 8.71 | < .001  
s80s20 | -3.11 | 0.74 | [-4.60, -1.63] | -4.18 | < .001

model\_performance(m1.1)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
-----------------------------------------------------------------  
676.482 | 676.802 | 683.590 | 0.185 | 0.175 | 16.854 | 17.072

We will learn more about the inferential statistics included in these model results (the *standard error*, the 95% *confidence interval*, the *t*-value and the *p*-value) later, once we are more confident with fitting and visualising regression models and interpreting the coefficients. For now, it’s enough to note that they are all related statistics testing the level of confidence that we can have in the generalisability of our results given the characteristics of our sample.

The statistics obtained from the model\_performance() function, on the other hand, describe the model overall and purport to aid with comparing the “performance” of different models. The R2 value (more precisely , R-squared) is called the *coefficient of determination* and measures how well a linear model predicts the outcome, representing the proportion of variation in the *response* variable that is predicted by the *explanatory* variables; in our case that’s 0.185 or 18.5%. That’s a reasonable proportion, but not nearly close to explaining all the variation in levels of trust, highlighting that there are various other factors also at play apart from inequality, both at the national and the individual levels.

## Standardizing the modelled variables

As mentioned earlier, *centring* our variables around the value “0” so that regardless of their measurement scale the value “0” becomes a meaningful number can be useful and help interpretation. Furthermore, if we constrain our *descriptor* variables to the same scale, then they become more easily comparable. Of course, we only have one *descriptor* variable at the moment, but that’s always just an initial phase in a real analysis project. The way we can constrain variables to a comparable scale is through *standardization*. A standardized variable (sometimes called a *z-score* or a standard score) is a variable that has been rescaled to have a mean of zero and a standard deviation of one.

To standardise a numeric variable, we can use the standardize() function. Let’s create a new standardised version of the s80s20 variable and save it as a new variable in the dataset:

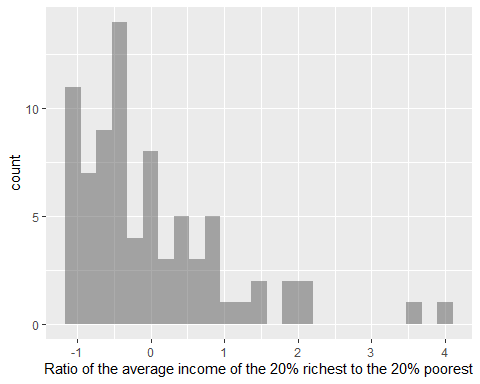
inequality <- inequality |>   
 data\_modify(s80s20\_std = standardize(s80s20))

We can plot the new variable:

describe\_distribution(inequality, s80s20\_std)

Variable | Mean | SD | IQR | Range | Skewness | Kurtosis | n | n\_Missing  
----------------------------------------------------------------------------------------  
s80s20\_std | 9.13e-17 | 1 | 1.07 | [-1.15, 3.90] | 1.67 | 3.50 | 79 | 10

gf\_histogram( ~ s80s20\_std, data = inequality)



Wee see that the *mean* of the new variable is very close to 0 (**9.13e-17** is just scientific notation for a very small and therefore very long number; it tells us to move the decimal point 17 places to the left to obtain the number), and the standard deviation is precisely 1.

If we were to refit the regression model with this variable, we would get:

lm(trust\_pct ~ s80s20\_std, data = inequality) |>   
 model\_parameters()

Parameter | Coefficient | SE | 95% CI | t(77) | p  
-------------------------------------------------------------------  
(Intercept) | 25.13 | 1.92 | [ 21.30, 28.95] | 13.08 | < .001  
s80s20 std | -8.09 | 1.93 | [-11.94, -4.24] | -4.18 | < .001

In this case, we find that the average value of *trust* for a country with an *average* (0) level of *inequality* is 25.13 (Intercept), while countries with a 1 standard deviation higher inequality than the average are expected to have a *trust* score that is 8.09 units (percentage points) lower than the average. Although the units of measurement have changed, the actual proportions of the effects we observe have not changed. We have made the intercept value meaningful, but the units by which we measure *inequality* are no longer the same. A *standard deviation* is a much larger unit than the units of the original scale, so a 1-unit change in inequality now translates into a larger effect.

We could also standardize the *response* variable in the same way and refit the model with all the modelled variables in standardized format:

inequality <- inequality |>   
 data\_modify(trust\_std = standardize(trust\_pct))  
  
m1.2 <- lm(trust\_std ~ s80s20\_std, data = inequality)  
  
model\_parameters(m1.2)

Parameter | Coefficient | SE | 95% CI | t(77) | p  
------------------------------------------------------------------  
(Intercept) | -0.03 | 0.10 | [-0.24, 0.17] | -0.31 | 0.754   
s80s20 std | -0.44 | 0.11 | [-0.65, -0.23] | -4.18 | < .001

The interesting thing about this output is that the coefficient we get for s80s20\_std is the same (within rounding error) as the *correlation coefficient* (r) between s80s20 and trust\_pct. To check the correlation between two numeric variables, we can either use the base-R function cor.test():

cor.test(inequality$trust\_pct, inequality$s80s20)

Pearson's product-moment correlation  
  
data: inequality$trust\_pct and inequality$s80s20  
t = -4.184, df = 77, p-value = 7.527e-05  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
 -0.5948878 -0.2312889  
sample estimates:  
 cor   
-0.430389

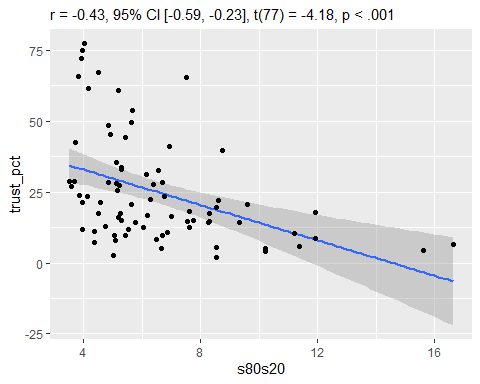
Or the cor\_test() function included in {easystats}:

cor\_test(inequality, "s80s20", "trust\_pct")

Parameter1 | Parameter2 | r | 95% CI | t(77) | p  
--------------------------------------------------------------------  
s80s20 | trust\_pct | -0.43 | [-0.59, -0.23] | -4.18 | < .001\*\*\*  
  
Observations: 79

The {easystats} ecosystem also has a convenience plot() function that creates quick plots from various summary functions, including cor\_test():

plot(cor\_test(inequality, "s80s20", "trust\_pct"))



A correlation is nothing more (or less) than a bivariate regression with both variables standardized. Its coefficient is a measure of the strength and direction of the association between two numerical variables. In this case, we observe a negative correlation between *inequality* and *trust* of medium strength.

From the regression model we obtained a related statistics, the *coefficient of determination* (), which is the square of the correlation coefficient ($-0.4304^2 = $0.1852347).

# Exercise 2: Predicting *trust* from *inequality* and *urbanization*, while accounting for *region* effects

About 30 minutes

In this exercise we extend the model we fit earlier by introducing two new variables: urban\_pop\_pct (the percentage of the urban population), and Region, the world geographical region to which the country belongs, which, as we saw earlier in the scatter-plot, complicates the simple correlation between *inequality* and *trust*.

## On your own: Perform the appropriate descriptive statistics for the two new variables

# Univariate descriptive statistics

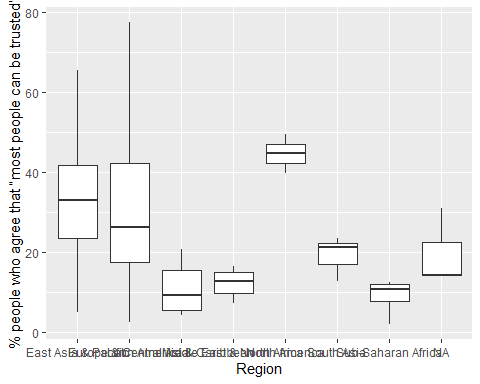
## Plot the relationship between *urbanisation* and *trust*, and between *urbanisation* and *inequality*

# Scatterplots

## Plot the relationship between *Region* and the other numerical variables

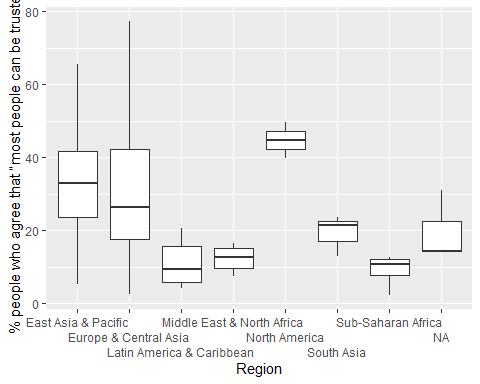
For this task, we need to introduce a new plot type: the *boxplot*. Box-plots are useful for visualising the relationship between a *categorical* and a *numeric* variable. They describe the distribution of the numeric variables in each category of the categorical variable:

gf\_boxplot(trust\_pct ~ Region, data = inequality)



To avoid the overlapping labels o the *x* axis, we can add some further specifications:

gf\_boxplot(trust\_pct ~ Region, data = inequality) +  
 scale\_x\_discrete(guide = guide\_axis(n.dodge=3))



Box-plots contain a lot of useful summary information about variables, and the interpretation of the shapes is the following:

|  |
| --- |
| box-plot |

**Create box-plots for the association between *region* and the other numeric variables**

# Box plots

To get the precise numeric values of the summary statistics captured in the box-plot, we can make a summary table using the means\_by\_group() function from the {easystats}:

means\_by\_group(inequality, "trust\_pct", "Region")

# Mean of % people who agree that “most people can be trusted” by Region  
  
Category | Mean | N | SD | 95% CI | p  
------------------------------------------------------------------------  
East Asia & Pacific | 33.10 | 15 | 17.86 | [24.65, 41.55] | 0.065  
Europe & Central Asia | 31.44 | 42 | 19.90 | [26.39, 36.48] | 0.062  
Latin America & Caribbean | 10.93 | 12 | 5.93 | [ 1.48, 20.37] | 0.062  
Middle East & North Africa | 12.42 | 8 | 3.42 | [ 0.85, 23.98] | 0.078  
North America | 44.62 | 2 | 6.90 | [21.49, 67.75] | 0.065  
South Asia | 19.25 | 3 | 5.58 | [ 0.36, 38.14] | 0.662  
Sub-Saharan Africa | 9.08 | 4 | 4.81 | [-7.28, 25.44] | 0.078  
Total | 25.94 | 86 | 18.63 | |   
  
Anova: R2=0.277; adj.R2=0.222; F=5.043; p<.001

What the function provides is effectively a type of regression results, as the statistics at the bottom of the table allude. Specifically, besides the mean value of trust in countries within each geographical grouping, we see statistics about the generalisability of the distributions and of model fit. The model that is being summarised is an Anova model (or *analysis of variance*) and its results are very similar to what we would obtain from a linear regression using the lm() function:

m1.3 <- lm(trust\_pct ~ Region , data = inequality)  
  
model\_parameters(m1.3)

Parameter | Coefficient | SE | 95% CI | t(79) | p  
--------------------------------------------------------------------------------------------  
(Intercept) | 33.10 | 4.24 | [ 24.65, 41.55] | 7.80 | < .001  
Region [Europe & Central Asia] | -1.66 | 4.94 | [-11.51, 8.18] | -0.34 | 0.737   
Region [Latin America & Caribbean] | -22.17 | 6.37 | [-34.84, -9.50] | -3.48 | < .001  
Region [Middle East & North Africa] | -20.68 | 7.20 | [-35.01, -6.36] | -2.87 | 0.005   
Region [North America] | 11.52 | 12.37 | [-13.11, 36.15] | 0.93 | 0.355   
Region [South Asia] | -13.85 | 10.40 | [-34.54, 6.84] | -1.33 | 0.187   
Region [Sub-Saharan Africa] | -24.02 | 9.25 | [-42.43, -5.61] | -2.60 | 0.011

Look at the regression results closely and compare it with the results from the Anova model in th previous table.

The logic of the linear regression model is different. As we know from the previous exercise, the coefficients represent a comparison between different values or levels of the *explanatory* variable. In the previous exercise, the comparison was to *a unit difference* in the level of *inequality*; here, R has recognised that Region is a categorical variable and has broken it down into a series of *indicator*/*dummy*/*binary* variables, one for each category of the region variable. Then, these individual *binary* indicator variables were intered into the regression model, effectively turning it into a multiple regression model with six predictors. The first category of the *Region* variable (“East Asia & Pacific”) appears to be missing. However, if we look closely, we notice that the mean value of “East Asia & Pacific” that we got in the previous summary table and the value of the *Intercept* in the regression model are the same. That’s because in the linear regression model results the “East Asia & Pacific” category was “absorbed” into the intercept - it became the “reference category” to which all the other categories compare.

As in the models in the previous exercise, the *Intercept* is purely the *average* value of the *response* variable trust\_pct when the value of the *explanatory* variables(s) is **0**. In our case, if the values of each of the six regions listed in the model are equal to **0** (i.e. a country does *not* belong to them), then the *Intercept* shows the average *trust* value of the remaining region (“East Asia & Pacific”). The other coefficients in the table are just the difference between the average *trust* in the reference category (“East Asia & Pacific”) and the other region. For instance the value of -1.66 associated with “Europe and Central Asia” means that the average *trust* of the countries in this region is , which is precisely the mean value that we saw in the means\_by\_group() table.

We can in fact remove the *Intercept* from the model by adding a column of **0** to the regression model like so:

m1.4 <- lm(trust\_pct ~ Region + 0 , data = inequality)  
  
model\_parameters(m1.4)

Parameter | Coefficient | SE | 95% CI | t(79) | p  
-------------------------------------------------------------------------------------------  
Region [East Asia & Pacific] | 33.10 | 4.24 | [24.65, 41.55] | 7.80 | < .001  
Region [Europe & Central Asia] | 31.44 | 2.54 | [26.39, 36.48] | 12.39 | < .001  
Region [Latin America & Caribbean] | 10.93 | 4.74 | [ 1.48, 20.37] | 2.30 | 0.024   
Region [Middle East & North Africa] | 12.42 | 5.81 | [ 0.85, 23.98] | 2.14 | 0.036   
Region [North America] | 44.62 | 11.62 | [21.49, 67.75] | 3.84 | < .001  
Region [South Asia] | 19.25 | 9.49 | [ 0.36, 38.14] | 2.03 | 0.046   
Region [Sub-Saharan Africa] | 9.08 | 8.22 | [-7.28, 25.44] | 1.10 | 0.273

## Build a multiple linear regression model

The aim of building a multiple regression model is to improve predictions of the *response* variable by taking account of the information provided by further variables. We can expand model m1.3 by including the *urbanisation* and *inequality* variables as well. Because at this stage we are still assuming an “additive” relationship, we can include further *explanatory* variables using the + sign:

m1.5 <- lm(trust\_pct ~ s80s20 + urban\_pop\_pct + Region , data = inequality)  
  
model\_parameters(m1.5)

Parameter | Coefficient | SE | 95% CI | t(69) | p  
---------------------------------------------------------------------------------------------  
(Intercept) | 12.99 | 10.77 | [ -8.49, 34.48] | 1.21 | 0.232   
s80s20 | -2.85 | 1.03 | [ -4.91, -0.79] | -2.76 | 0.007   
urban pop pct | 0.56 | 0.11 | [ 0.34, 0.77] | 5.15 | < .001  
Region [Europe & Central Asia] | -3.88 | 4.82 | [-13.49, 5.73] | -0.81 | 0.423   
Region [Latin America & Caribbean] | -12.57 | 8.15 | [-28.83, 3.69] | -1.54 | 0.128   
Region [Middle East & North Africa] | -23.70 | 6.74 | [-37.15, -10.25] | -3.52 | < .001  
Region [North America] | 6.57 | 10.82 | [-15.01, 28.15] | 0.61 | 0.546   
Region [South Asia] | -1.74 | 9.59 | [-20.87, 17.39] | -0.18 | 0.856   
Region [Sub-Saharan Africa] | -2.74 | 8.60 | [-19.91, 14.42] | -0.32 | 0.751

model\_performance(m1.5)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
-----------------------------------------------------------------  
640.261 | 643.545 | 663.828 | 0.527 | 0.472 | 12.898 | 13.714

The interpretation of the coefficients is similar to the one in the previous model. The *Intercept* still represents the average *trust* value when the *explanatory* variables hold the value **0**. In this case, it represents the average trust of a country in “East Asia and the Pacific” with 0% of its population living in cities and with an income quintile share ratio of “0”. As with the first model, such a country cannot exist in reality, so the *Intercept* value should not be interpreted. However, the slope values of the other coefficients can be interpreted as the difference in the *trust* value when the variable takes on a value that is one unit higher (in the scale on which the variable is measured), while taking into account the effect of the other variables included in the model.

We find that the *inequality* variable still has a negative and notable association with *trust* even after accounting for differences in urbanization and world region.

# Exercise 3: Are anti-immigrant attitudes associated with lower social trust?

About 30 minutes

In this exercise you can attempt to address the research question in the title on your own, using the steps and modelling techniques practiced in the previous exercises.

The “application” reading by (Mitchell 2021) has addressed a similar question, and you can use that article as background information. To address the question, you will be using data from the latest round of the **European Social Survey**. You can download the **ESS10** dataset and check the variable codebook here: https://soc2069.chrismoreh.com/data/data\_main

Your task is to:

* download the data and import it to R
* using the variable codebook, identify potentially useful variables. You need to identify a variable relating to *generalised social trust*, and a variable of your choice that measures attitudes towards immigration
* describe your variables individually and in relationship to each other
* fit and interpret the results from a linear regression model

# References:

Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2020. *Regression and other stories*. Cambridge: Cambridge University Press.

Mitchell, Jeffrey. 2021. “Social Trust and Anti-immigrant Attitudes in Europe: A Longitudinal Multi-Level Analysis.” *Frontiers in Sociology* 6 (April): 604884. <https://doi.org/10.3389/fsoc.2021.604884>.

Wilkinson, Richard G., and Kate Pickett. 2010. *The Spirit Level: Why Greater Equality Makes Societies Stronger*. New York: Bloomsbury Press.