Intro to R

## Why use R?

* Open source and free
  + R is available for free, making it accessible to everyone
  + A large, active community contributes to continuous improvement and knowledge sharing
  + Strong in statistical analysis
* R has a wide array of packages dedicated to statistical analysis
  + R is great at producing complex plots
  + R has specialised tools for various fields, such as econometrics and randomisation for research and evaluation
* Reproducibility
  + Script based workflow: R uses scripts for analysis, which can be shared and re-executed to reproduce results
  + R works well with Git for version control, helping track changes and collaborate efficiently
  + Tools like Quarto allow for the integration of code, results, and text into a single document, ensuring analyses are transparent and reproducible

## Workflow

* RStudio: An IDE (integrated development environment) for R
  + RStudio provides a user friendly interface
* Working directories
  + The folder where R looks for files and saves output
  + Avoid using relative paths - use R projects
* Creating an R project
  + R Projects help organise all related files, scripts, and data in a single directory
* Scripts
  + Scripts are written in .R files, where you can save commands and code for later use
  + You can run line-by-line or in chunks, making it easy to test and debug your scripts
  + Use comments within scripts to explain the purpose of code sections, improving readability and collaboration.
* Packages
  + Packages can be installed using install.packages()
  + Once installed packages are loaded into the R session using library()
  + Make sure to install and load tidyverse
    - install.packages("tidyverse")
    - library(tidyverse)

## Try it!

Download and load packages. You only need to install it once.

#install.packages("tidyverse")  
#install.packages("devtools")   
#devtools::install\_github("jakubkuzilek/oulad")  
  
  
library(tidyverse)

Warning: package 'tidyverse' was built under R version 4.4.1

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(oulad)

Load the data.

data("student", package = "oulad")

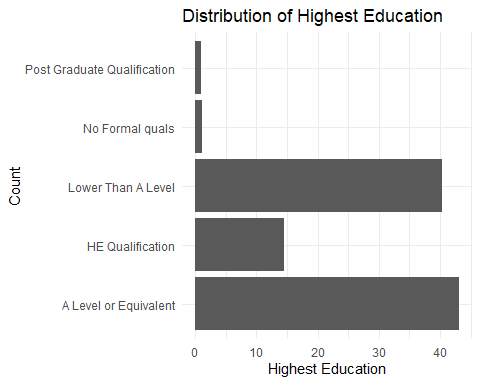
Create a count.

student\_summary <- student %>%  
 count(highest\_education) %>%  
 mutate(percentage = round(n / sum(n) \* 100, 2)) %>%  
 arrange(desc(n))  
  
student\_summary

highest\_education n percentage  
1 A Level or Equivalent 14045 43.09  
2 Lower Than A Level 13158 40.37  
3 HE Qualification 4730 14.51  
4 No Formal quals 347 1.06  
5 Post Graduate Qualification 313 0.96

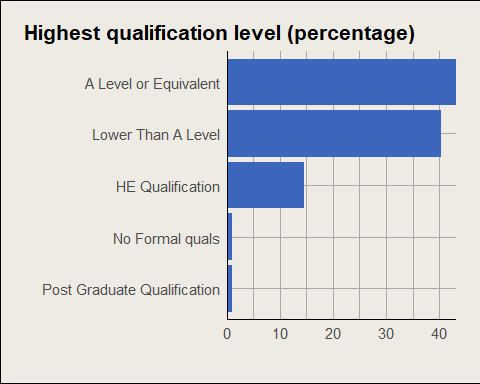
Make a basic chart.

ggplot(student\_summary, aes(x = percentage, y = highest\_education)) +  
 geom\_bar(stat = "identity") +  
 labs(  
 title = "Distribution of Highest Education",  
 x = "Highest Education",  
 y = "Count"  
 ) +  
 theme\_minimal()



Customise it

ggplot(student\_summary, aes(x = percentage, y = fct\_reorder(highest\_education, percentage))) +  
 geom\_bar(stat = "identity", fill = "#3b66bc") +  
 labs(  
 title = "Highest qualification level (percentage)",  
 x = "",  
 y = ""  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title.position = "plot",  
 plot.title = element\_text(size = 16, face = "bold"),  
 plot.subtitle = element\_text(size = 12),  
 plot.caption.position = "plot",  
 plot.caption = element\_text(hjust = 0, size = 8, face = "italic"),  
 panel.grid.major = element\_line(colour = "darkgrey"),  
 panel.grid.minor = element\_line(colour = "darkgrey"),  
 plot.background = element\_rect(fill = "#EDEBE3"),  
 plot.margin = margin(0.25, 0.25, 0.25, 0.25, "in"),  
 axis.text.x = element\_text(size = 11),  
 axis.text.y = element\_text(size = 11),  
 axis.line = element\_line(colour = "black", linewidth = 0.5),  
 legend.position = "none"  
 ) +  
 scale\_x\_continuous(expand = c(0, 0))



## Reporting with Quarto

* Case study: [TASO Technical Guide](https://taso-he.github.io/technicalguide/)

## Resources for further learning

* [R for Data Science](https://r4ds.hadley.nz/intro)
* [TASO’s coding good practice](https://taso-he.github.io/technicalguide/coding-good-practice/)
* [TASO data visualisation style guide](https://taso-he.github.io/technicalguide/data-vis/)
* [Productive R Workflow](https://www.productive-r-workflow.com/) (paid for course)