Excel Programming and Automation

Jonas Moss

12/12/22

Table of contents

Introduction

Welcome to the course ELE 3915 Excel Programming and Automation. You're about to get your hands dirty! This course is about being able to do cool things Excel and TypeScript. It should give you the confidence to assert that you can program in Excel. And it gives you an introduction to TypeScript, a popular general-purpose programming language used in Microsoft Office on the web. The slides can be found on Github, in the `slides` directory.

If you need to get in contact with me, please send an e-mail to `jonas.moss@bi.no`. I do not check It's learning often.

> **Warning**
>
> This webpage is being continuously updated during. Most of the chapters are not finished.

Rough structure

The course is split into into four parts.

1. **Lecture 1–3**. Basic Excel usage, including formulas such as `SUM` ,`COUNTIFS`, `INDEX`, and `MATCH`. We focus on general mathematical formulas and data manipulation, in addition to simple formatting. Roughly speaking, these lectures cover the sort of Excel you should expect to see in lecacy Excel sheets. Most of the curriculum is covered on the excellent webpage ExcelJet.
2. **Lecture 4–6**. These lectures covers modern Excel formulas using dynamic arrays, `LAMBDA` and `LET`. Many of these features are just a couple of years old (as of 2023). I would not expect most employers to know about them. These features can do quite a bit of what Visual Basic for Applications (VBA) did in legacy worksheets. Programming using `LAMBDA` and `LET` is difficult, so make sure you take the exercises seriously. Again, most of the curriculum is covered on ExcelJet.
3. **Lecture 7–9**. Covers power query, power pivot, and DAX. These are modern tools for data visualization, data cleaning, and data manipulation. These tools are likely to form the core of your automation in Excel. They are widely regarded as the most important tools in the Excel analysts toolbox.

4. **Lecture 10–14**. These lectures contain an introduction to TypeScript and its application in Excel. Take note that we assume no previous exposure to programming in this course. TypeScript is a *typed* modification of JavaScript, the dominant language in web development, and has application far beyond Excel. The lectures will focus on programming in TypeScript in general, and the emphasis is on exercises.

A note on difficulty

Most of the course is about programming. Students often find programming hard. **Don't expect to be able solve every exercise in 5 minutes!** Solving programming exercises often take a long time, and you need to persevere.

Be aware that the difficulty of this course is uneven. For instance, lectures on `LAMBDA` and `LET` are harder than the first three lectures, and the lectures on power query are likely to be easier. Do not think this course will be a walk in the park.

To become a decent programmer it's a good idea to

1. Do a lot of exercises.
2. Spend at least 20 minutes on each exercise before you give up. You need to think really hard. Don't expect to be able to solve the problem without making an effort.
3. Do the exercise yourself after you have looked at the solution! Close the window and do it from memory. It's also a good idea to revisit the same exercise later on, e.g. the next day, to make sure you're able to do it.
4. Tinker around, either modifying exercises yourself, or with your own ideas. If your tinkering leads to something cool, tell me! Use Kaggle to download data sets to tinker with and Mockaroo to generate fake but plausible-looking data sets.

Do not to spend an inordinate amount of time on an exercise before you check the solution. If you have spent 1 hour on an exercise and haven't gotten anywhere, it might be smart save yourself some time and look at the solution.

Moreover, be aware that programming is often *extremely frustrating*. It's like talking to someone who just simply refuses to understand what you're saying, no matter how many times you repeat yourself. **It's normal and expected to feel frustrated**!

There are many tips online about learning to program, e.g., this collection of tips. But it mostly boils down to spending a lot of time solving problems.

About this site

Curious how this site was made? It is written using Quarto books.

1 Excel basics (i): Introduction to formulas

Last updated: 03.01.2023

1.1 Curriculum

- Basics of Excel, including cells, active cells, and ranges. Worksheets, basic formatting, data types and blank cells.
- Understanding what formulas and functions are, including copying of formulas using relative references.
- Reading the signature and short documentation of functions inside Excel.
- Using flash fill.
- Using the operators `+`,`-`,`/`, and `*`.
- A couple of keyboard shortcuts.
  - Press F2 to edit a cell and see its dependencies.
  - Press escape to exit editing a formula.
  - Press tab to auto-complete a formula.
  - Press SHIFT to add contiguous cells to a selection.
  - Press CTRL+DOWN to go to the last non-blank cell in a contiguous column. And CTRL+UP to go to the first column-wise, CTRL+LEFT-ARROW to go to the first row-wise, and CTRL+RIGHT-ARROW to go to the last row-wise.
  - Hold CTRL to add cells to a selection.
  - Press F4 to taggle absolute references.
- Logical values; the functions `AND`, `OR`, `NOT`, and the operators `=`, `<`, `>`, `<=` and `>=`.
- Overview of error messages.
- Knowledge of the basic datatypes of Excel (number; text; logical; error; array), including how to use the `TYPE` function.

Basic functions using one range as argument.

Name	Description
SUM	Sum all numbers in a range.
PRODUCT	The product of all numbers in a range.
MAX	The maximum of numbers in a range.
MIN	The minimum of numbers in a range.
COUNT/COUNTA/COUNTBLANK	Count the number of cells in a range that contain numbers (COUNT), are non-empty (COUNTA), or empty (COUNTBLANK).
ROWS / COLS	Counts the number of rows / columns in a reference.
AVERAGE	The average of the numbers in a range. Empty cells are ignored.
MEDIAN	The median of the numbers in a range.
STDEV	The standard deviation of the numbers in a range. Empty cells are ignored.
COUNTIF	Counts cells satisfying a criterion.

1.2 Slides and sheets used in the lecture

Remember to look at the solutions only after giving the exercises a serious attempt. Solve the exercises yourself after looking at the solution.

The lecture slides are here. The Excel sheets used in the lecture, before being filled in, here. The lecture notes after being filled in are here.

Here are the exercises; the solutions can be found here.

1.3 Recommended resources

There are many excellent video resources for Excel online. The content of this lecture is pretty standard stuff, and there are probably 100s of Youtube videoes covering essentially the same content. For instance, the formulas covered in this lecture are also covered by Kevin Stratvert here, but he goes a little further, covering harder formulas too. Leila Gharani introduces formulas here.

There are many introductions to flash fill too, e.g. this one.

The site Excel Exercises provides some decent exercises, but goes beyond the content of this lecture.

There is a staggering number of shortcuts in Excel, see e.g. here. It's easy to get overwhelmed by shortcuts, so make sure you don't try to learn too many at once though!

2 Excel basics (ii)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

2.1 Curriculum

2.1.1 Absolute and relative references.

- Know the difference between absolute, mixed, and relative references and when to use which.
- Use F4 to cycle between the different kinds of references.

2.1.2 Conditionals in Excel.

- Conditional formulas
  - `IF` function
  - Oldskool nested `IF`
  - The better `IFS` function
- The `SUMIF`/`SUMIFS`; `COUNTIF`/`COUNTIFS`; `MINIFS`/`MAXIFS` functions.

2.1.3 Working with text

- Get an overview of the text manipulation functions in Excel and how to apply them.
- Be able to solve basic text manipulation tasks.
- Working with strings. Using `&` for concatenation and F9 for replacing formulas with values.
- Not all useful text manipulation tools are built-in though. For instance, `WORDCOUNT`, counting the number of words in a text, does not exist. We will revisit text manipulation later on.

- We will revisit text manipulation later on.
- Some of the functions are oldskool. You might see them used, but modern alternatives are better.
  - This is arguably the case for `LEFT`; `MID`, and `RIGHT`, `FIND` and `SEARCH`.

Name	Description
`TRIM`	Remove extra spaces from text.
`TEXTBEFORE` / `TEXTAFTER`	Extract text before / after delimiter.
`EXACT`	Compare two text strings, taking case into account.
`SUBSTITUTE`	Replace text based on content.
`LOWER`/`UPPER`	Transform text to lower/upper case.
`PROPER`	Capitalize first letter of each word in text.
`CONCAT` / `TEXTJOIN`	Join text values with(out) a delimiter.
`LEN`	Get the length of the text
`LEFT` / `MID` / `RIGHT`	Extract text from the left/middle/right of a string
`FIND`	Get location substring in a string.
`REPLACE`	Replace text based on location.

2.2 Exercises

2.3 Recommended resources

https://www.youtube.com/watch?v=thvE8Eg-Pqg&ab_channel=Chandoo

3 Excel basics (iii)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

3.1 Curriculum

3.1.1 Part 1

- Data tables.
- Data validation.
- Conditional formatting.
- Rounding of numbers: `ROUND`; `FLOOR`; `CEILING`; etc.
- Ranking and ordering: `RANK`/`RANKEQ` function, the `SMALL` and `LARGE` functions.
- Functions of multiple ranges: `SUMPRODUCT`.

3.1.2 Part 2

- The `INDEX`, `ROWS`, and `COLUMNS` functions.
- The `MATCH` function;
- Using `INDEX` together with `MATCH`.
- The old school look up functions `VLOOKUP`, `HLOOKUP`, `LOOKUP`.
- The modern look up function: `XLOOKUP`.
- The modern match function: `XMATCH`.

3.2 Exercises

3.3 Recommended resources

4 Lambdas and dynamic arrays

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

4.1 Curriculum

- How to define your own functions using `LAMBDA` and why it is useful.
  1. It makes readers of your code understand your intention. Including yourself in the future!
  2. Functions makes it easier to do hard things, as you can split up complicated formulas into multiple parts.
- Short introduction to the Advanced Formula Environment
- What a dynamic array is and what the `#SPILL!` error means.
- Some important functions:
  - `FILTER`
  - `SORT`, `SORTBY`
  - `UNIQUE`
  - `TEXTSPLIT`

4.2 Exercises

4.3 Recommended resources

5 User-made functions with LAMBDA

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

5.1 Curriculum

Here is an example of a complex function.

```
=LET(
    tab; FILTER(vgsales; (vgsales[Genre] = genre) * (vgsales[Platform] = platform));
    rows; {7; 8; 9; 10};
    regions; {"NA"; "EU"; "JP"; "Other"};
    VSTACK(
        HSTACK(""; TRANSPOSE(Years));
        HSTACK(
            regions;
            MAKEARRAY(
                ROWS(rows);
                ROWS(Years);
                LAMBDA(row_index; year_index;
                LET(
                    row;index(rows;row_index);
                    year;index(Years;year_index);
                    IFERROR(SUM(FILTER(CHOOSECOLS(tab; row); CHOOSECOLS(tab; 4) = year));
                )
            )
        )
    )
)
```

5.2 Exercises

5.3 Recommended resources

6 LET, LAMBDA, and dynamic arrays

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

6.1 Curriculum

Excel is equipped with many functions for handling dynamic arrays and `LAMBDA` functions. The most important are `FILTER`, `MAP` (and its cousins `BYCOL`, `BYROW`, and `MAKEARRAY`), and `REDUCE`.

To use `FILTER`, `MAP`, and `reduce`, we will need convenience functions such as

6.2 Exercises

6.2.1 Utility functions

We have discussed

Recall that `FILTER` does not take the same sort of arguments as e.g. `MAP`. Its second argument is an array of truth values, not a `LAMBDA` function. That is OK, but it's inconsistent with most ofther functions of dynamic arrays. Make a function

- `FILTER_(array; lambda)` Returns the array consisting of elements from `array` where `lambda(x)` is `TRUE`.
- `TAKEWHILE(array; lambda)` Takes elements from `array` until the `LAMBDA` function `lambda` returns false.
- `MAPWHILE(array;mapper;predicate)` Maps `mapper` onto array untile the `predicate` function returns `FALSE`.
- `FILTERBY(out;filter_array;lambda)` Returns the array consisting of elements from `out` where `lambda(x)` evaluates to `TRUE`.
- `ENUMERATE(array)` Takes a one-dimensional array (a column) merges it with `SEQUENCE(R)`, yielding a two-dimensional array.

6.3 Recommended resources

7 Power query (i)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

7.1 Curriculum

7.2 Exercises

7.3 Recommended resources

8 Power query (ii)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

8.1 Curriculum

8.2 Exercises

8.3 Recommended resources

9 Power pivot and DAX

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

9.1 Curriculum

9.2 Exercises

9.3 Recommended resources

10 Typescript (i)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

10.1 Curriculum

- **Primitive JavaScript values.** A list can be found here.
  - `Undefined` and `Null`.
  - Core primitives: `Boolean`, `Number`, and `String`.
  - Specialized: `BigInt` and `Symbol`.
- **Arrays:** Unnamed collection of objects.
- **Functions:** Takes input vales and does something with them.

10.2 Exercises

10.3 Recommended resources

11 Typescript (ii)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

11.1 Curriculum

11.2 Exercises

11.3 Recommended resources

12 Typescript (iii)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

12.1 Curriculum

12.2 Exercises

12.3 Recommended resources

13 Typescript (iv)

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

13.1 Curriculum

13.2 Exercises

13.3 Recommended resources

14 TypeScript in Excel

> **!** Important
>
> This page is ***not finished***, but might contain notes from the course developers. The curriculum, exercises, and recommended resources listed on this page is subject to change.

14.1 Curriculum

14.2 Exercises

14.3 Solutions to exercises

14.4 Recommended resources