# main

October 7, 2025

```
In [2]: ### Global imports
        import dill
        from cse6040_devkit import plugins, utils

        utils.add_from_file('df_level_apply_checker', plugins)
        utils.add_from_file('melt_blocker', plugins)
        utils.add_from_file('pivot_table_blocker', plugins)
```

```
cse6040_devkit.plugins
cse6040_devkit.plugins
cse6040_devkit.plugins
```

```
In [3]: # Some modules you'll need in this part
        import pandas as pd
        import numpy as np
        from io import StringIO
        from IPython.display import display
        import re
```

## 1 Part 0: Getting the data

Before beginning, you'll need to download several files containing the data for the exercises below.

**Ungraded**. Run the code cell below to download the data.

```
In [4]: irises = pd.read_csv('resource/asnlib/publicdata/iris.csv')
        table1 = pd.read_csv('resource/asnlib/publicdata/table1.csv')
        table2 = pd.read_csv('resource/asnlib/publicdata/table2.csv')
        table3 = pd.read_csv('resource/asnlib/publicdata/table3.csv')
        table4a = pd.read_csv('resource/asnlib/publicdata/table4a.csv')
        table4b = pd.read_csv('resource/asnlib/publicdata/table4b.csv')
        who = pd.read_csv('resource/asnlib/publicdata/who.csv')
        who2 = pd.read_csv('resource/asnlib/publicdata/who2_soln.csv')

        print("\n(All data appear to be ready.)")
```

```
(All data appear to be ready.)
```

## 2 Part 1: Tidy data

The overall topic for this lab is what we'll refer to as representing data *relationally*. The topic of this part is a specific type of relational representation sometimes referred to as the *tidy* (as opposed to *untidy* or *messy*) form. The concept of tidy data was developed by Hadley Wickham, a statistician and R programming maestro. Much of this lab is based on his tutorial materials (see below).

If you know SQL, then you are already familiar with relational data representations. However, we might discuss it a little differently from the way you may have encountered the subject previously. The main reason is our overall goal in the class: to build data *analysis* pipelines. If our end goal is analysis, then we often want to extract or prepare data in a way that makes analysis easier.

You may find it helpful to also refer to the original materials on which this lab is based:

- Wickham's R tutorial on making data tidy: http://r4ds.had.co.nz/tidy-data.html
- The slides from a talk by Wickham on the concept: http://vita.had.co.nz/papers/tidy-data-pres.pdf

---

### 2.1 What is tidy data?

To build your intuition, consider the following data set collected from a survey or study.

**Representation 1.** Two-way contigency table.

| | Pregnant | Not pregnant |
|---|---|---|
| **Male** | 0 | 5 |
| **Female** | 1 | 4 |

**Representation 2.** Observation list or "data frame."

| Gender | Pregnant | Count |
|---|---|---|
| Male | Yes | 0 |
| Male | No | 5 |
| Female | Yes | 1 |
| Female | No | 4 |

These are two entirely equivalent ways of representing the same data. However, each may be suited to a particular task.

For instance, Representation 1 is a typical input format for statistical routines that implement Pearson's $\chi^2$-test, which can check for independence between factors. (Are gender and pregnancy status independent?) By contrast, Representation 2 might be better suited to regression. (Can you predict relative counts from gender and pregnancy status?)

While Representation 1 has its uses, Wickham argues that Representation 2 is often the cleaner and more general way to supply data to a wide variety of statistical analysis and visualization tasks. He refers to Representation 2 as *tidy* and Representation 1 as *untidy* or *messy*.

The term "messy" is, as Wickham states, not intended to be perjorative since "messy"