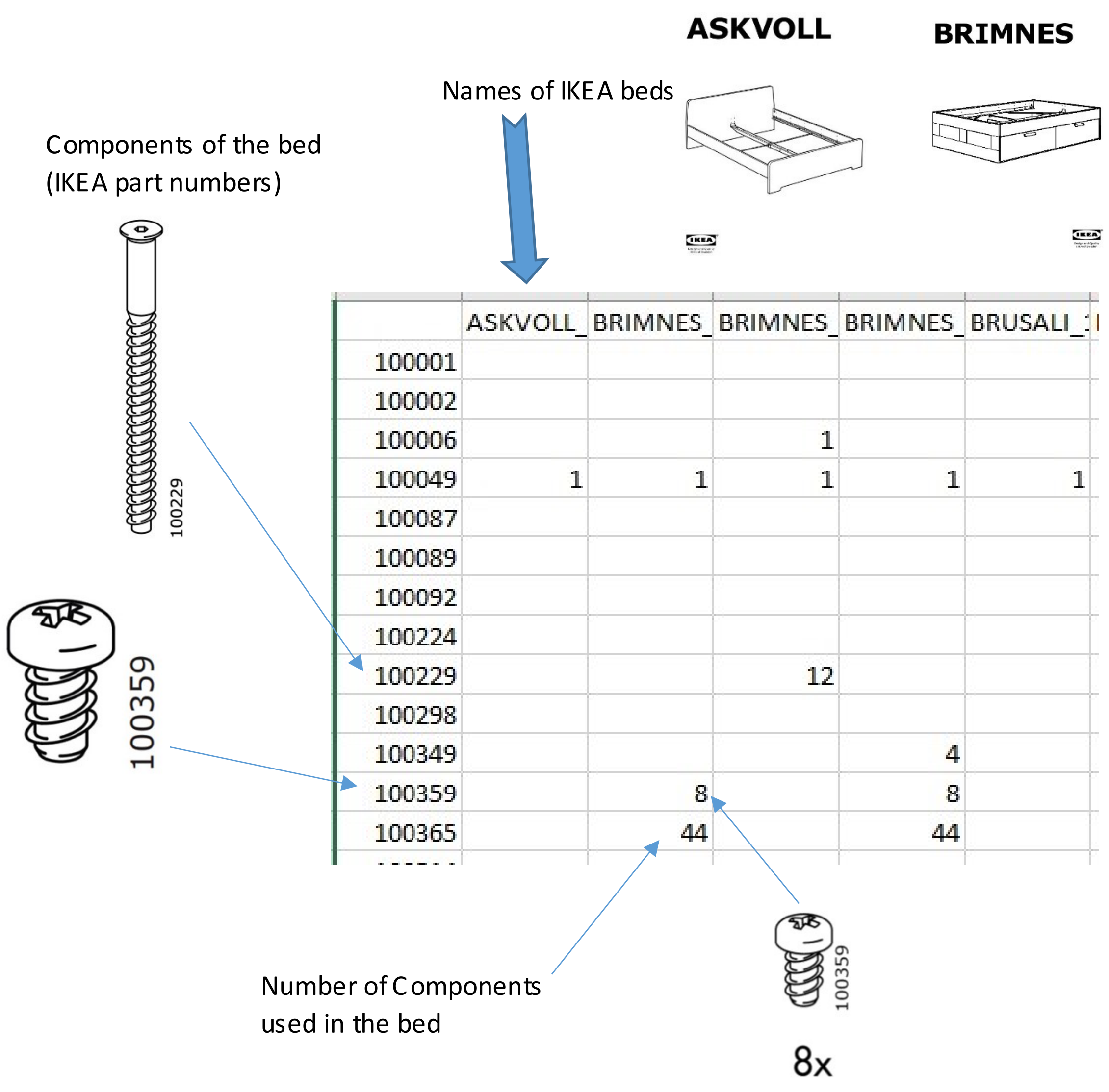
Bed Matrix Format



	ASKVOLL_	BRIMNES_	BRIMNES_	BRIMNES_	BRUSALI_
100001					
100002					
100006			1		
100049	1	1	1	1	1
100087					
100089					
100092					
100224					
100229			12		
100298					
100349				4	
100359		8		8	
100365		44		44	

The CSV files need cleaned to create a uniform format of text, part numbers, rows and columns (including the removal of spurious data). After cleaning some “carpentry” *might* be required to get it into a format that enables the analysis of your choice.

You might, for example, want to create a NumPy Array that just records which components are in which bed (i.e. disregarding the quantity):

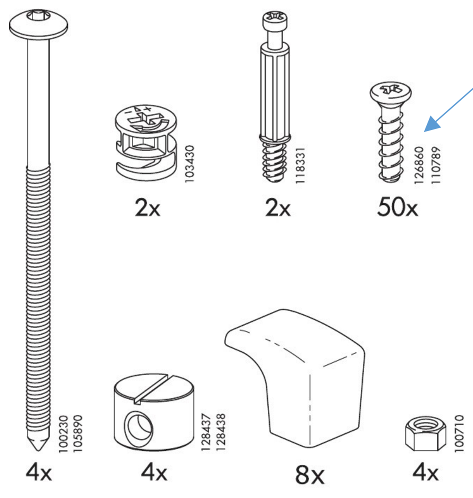
[[ASKVOLL, 100049, 100359, 102267, 105163,113453, 114670, 117327, 122628,124401, 124402, 139301, 139430, ‘126860/110789’] [BRIMNES, 100049, …..]…….]

Or perhaps, you want to record the total number of components in each bed (i.e. disregarding their type):

[[ASKVOLL, 75], [BRIMNES_179_15, 277], [BRIMNES_199_19, …..] …..]

Note: There are many Pandas functions available to support this sort of manipulation (e.g. transpose). See the Pandas ‘cheat sheets’ on learn (Course Materials) for a quick overview. Load the data and experiment.

Special Case: Parts with more than one number



Sometimes the same physical part has two separate number.
There isn't a right, or a wrong, way of dealing with this.
You just have to decide a rule (ie procedure),
for parts with multiple numbers, and state it in your report.

You could, for example, divide up the string and just use the lowest number to represent the part.

```
s = "12345/98765"   # string with a / seperator
b = s.split('/')   # divide the string into a number of parts
print('b =', b)
print('b[0] =', b[0])
print('b[1] =', b[1])
a = b[0]
print('a =', a)
```

```
b = ['12345', '98765']
b[0] = 12345
b[1] = 98765
a = 12345
```