**Instructions to convert Excel (.xls and .xlsx) files into RDF files:**

1. Place the Excel2CSV.py file in the same directory as the the Excel file you wish to convert.

2. Open your terminal and navigate to that directory that contains the data and program.

e.g. bash$: cd ~/Desktop/data\_folder

3. Run the Excel2CVS.py program following either of the two usages below.

bash$:~/Desktop/data\_folder python Excel2CSV.py <excel\_filename.xls> <csv\_filename.csv>

bash$:~/Desktop/data\_folder python Excel2CSV.py <excel\_filename.xlsx> <csv\_filename.csv>

e.g. bash$:~/Desktop/data\_folder python Excel2CSV.py measured\_vals.xls csv\_measured\_vals.csv

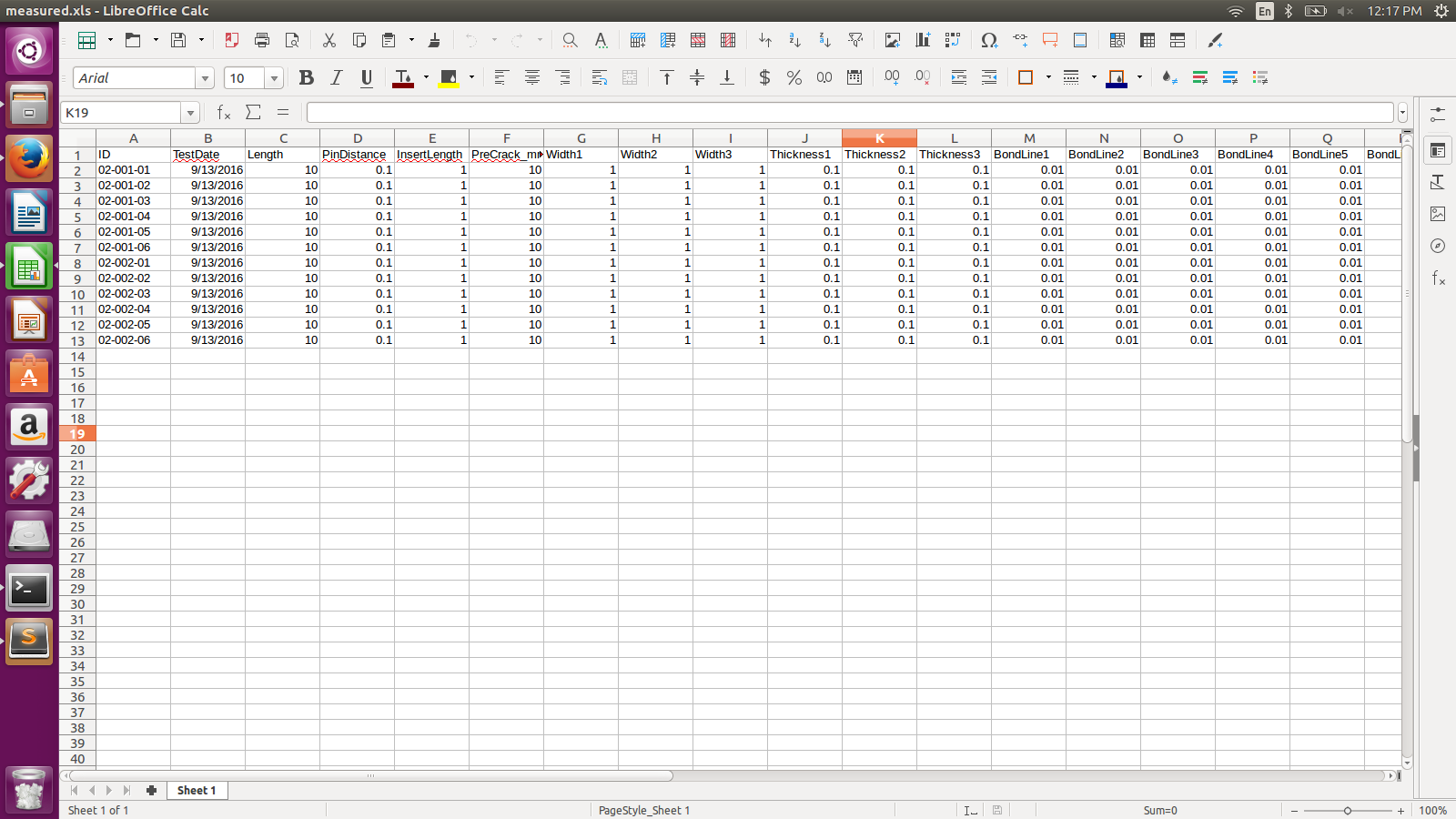
This will construct a CSV file from the Excel file in the same directory.

4. Run the csv2rdf4lod program using the instructions found on [github](https://github.com/timrdf/csv2rdf4lod-automation/wiki/conversion:Enhancement).

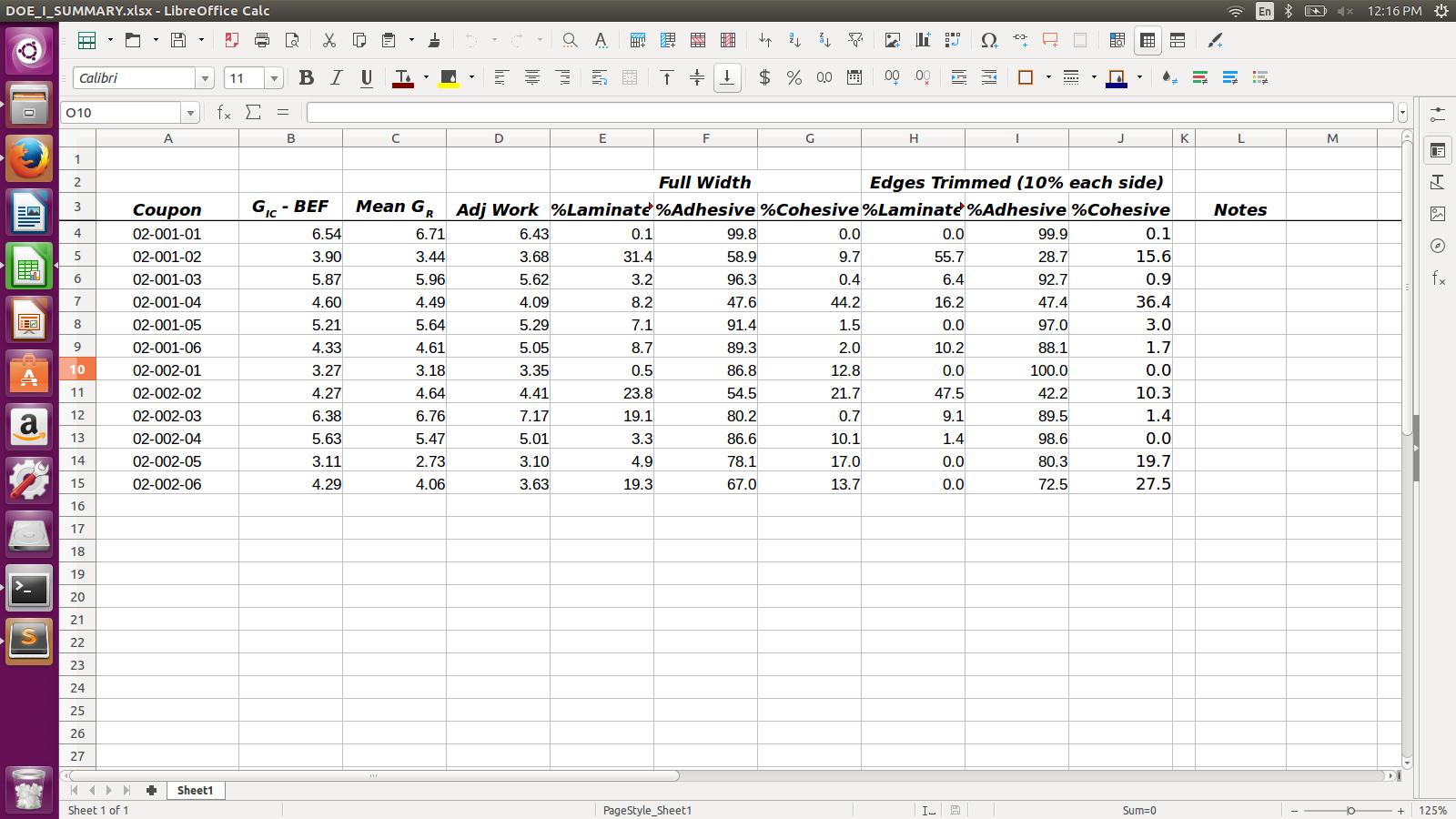
**Notes regarding the current program:**

1. This program currently only works with \*.xls and \*.xlsx file extensions.

2. Just like with typical CSV files, the csv2rdf4lod software only operates correctly if the headers start on row 1 of the file.

e.g.

is okay, but



is not okay. This is because the first row in the above picture has a blank row in row 1. It also has two merged cells that are not needed in the RDF file (at least I do not believe so). So even if the empty row was deleted, the RDF file would only consider “Full Width” and “Edges Trimmed (10% each side)” as the properties of the file, which is incorrect.

Therefore, in later iterations of the python file I created, I would only start copying data to the CSV file that results from it once it notices the entire row of headers (which is assumed to be the important one).