```
!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in /usr/local/lib/python3.10/di
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.10/
```

```
# required packages `!pip install openpyxl`
import openpyxl
```

```
# name of the excel workbook
xlsx_file_path = '/unicef_sowc.xlsx'
# load the excel spreadsheet (workbook)
workbook = openpyxl.load_workbook(xlsx_file_path)
# print to make sure it loaded - `sanity` test or `debug` test
print(workbook)
```

```
<openpyxl.workbook.workbook.Workbook object at 0x7ffaf894eda0>
```

```
# variable to hold the names of the sheets
sheet_names = workbook.sheetnames
# iterate through the sheet names and print them
print("Names of the sheets in the workbook:")
for sheet_name in sheet_names:
  print(sheet_name)
```

```
Names of the sheets in the workbook:
Data Notes
Table 9
```

```
# name of the sheet you want to access
sheet_name = 'Table 9' # expect an error
# access the specific sheet by name
sheet = workbook[sheet_name]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-13-154426151839> in <cell line: 4>()
      2 sheet_name = 'Table 9' # expect an error
      3 # access the specific sheet by name
----> 4 sheet = workbook[sheet_name]

/usr/local/lib/python3.10/dist-packages/openpyxl/workbook/workbook.py in __getitem__(self, key)
    285             if sheet.title == key:
    286                 return sheet
--> 287         raise KeyError("Worksheet {0} does not exist.".format(key))
    288
    289     def __delitem__(self, key):
```

Next steps: Explain error

```
# name of the sheet you want to access
sheet_name = 'Table 9 ' # fixed spacing
# access the specific sheet by name
sheet = workbook[sheet_name]
# print to make sure it loaded - `sanity` test or `debug` test
print(sheet)
```

```
<Worksheet "Table 9 ">
```

```
# show what methods are available
print(dir(sheet))
```

```
['BREAK_COLUMN', 'BREAK_NONE', 'BREAK_ROW', 'HeaderFooter', 'ORIENTATIO
```

```
# shows it is iterable (we can use a for loop)
print(sheet.rows)
```

```
<generator object Worksheet._cells_by_row at 0x7ffae2163df0>
```

```
# documentation on the `rows` method
help(sheet.rows)
```

```
Help on generator object:

_cells_by_row = class generator(object)
 |  Methods defined here:
 |
 |  __del__(...)
 |
 |  __getattribute__(self, name, /)
 |      Return getattr(self, name).
 |
 |  __iter__(self, /)
 |      Implement iter(self).
 |
 |  __next__(self, /)
 |      Implement next(self).
 |
 |  __repr__(self, /)
 |      Return repr(self).
 |
 |  close(...)
 |      close() -> raise GeneratorExit inside generator.
 |
 |  send(...)
 |      send(arg) -> send 'arg' into generator,
 |      return next yielded value or raise StopIteration.
 |
 |  throw(...)
 |      throw(value)
 |      throw(type[,value[,tb]])
 |
 |      Raise exception in generator, return next yielded value or rais
 |      StopIteration.
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  gi_code
 |
 |  gi_frame
 |
 |  gi_running
 |
 |  gi_yieldfrom
 |      object being iterated by yield from, or None
```

```
# raw data from the worksheet
```

```
# iterate over each row and cell, then print the values
for row in sheet.rows:
  for cell in row:
    print(cell.value, end='/t')
  print()
```

```
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tDEFINITIONS OF THE INDICATORS/tNone/tNone/tNone/tNone/tNone/tNone
None/tChild labour – Percentage of children 5–14 years old involved in
None/tChild marriage – Percentage of women 20–24 years old who were fir
None/tBirth registration – Percentage of children less than 5 years old
None/tFemale genital mutilation/cutting (FGM/C) – (a) Women: percentage
None/tJustification of wife beating – Percentage of women and men 15–49
None/tViolent discipline – Percentage of children 2–14 years old who ex
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tMAIN DATA SOURCES/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone
None/tChild labour – Multiple Indicator Cluster Surveys (MICS), Demogra
None/tChild marriage – MICS, DHS and other national surveys./tNone/tNon
None/tBirth registration – MICS, DHS, other national household surveys,
None/tFemale genital mutilation/cutting – MICS, DHS and other national
None/tJustification of wife beating – MICS, DHS and other national surv
None/tViolent discipline – MICS, DHS and other national surveys./tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tDÉFINITIONS DES INDICATEURS/tNone/tNone/tNone/tNone/tNone/t
None/tNone/tTravail des enfants – Pourcentage d’enfants âgés de 5 à 14
None/tNone/tMariage d’enfants – Pourcentage de femmes âgées de 20 à 24
None/tNone/tEnregistrement des naissances – Pourcentage d’enfants de mo
None/tNone/tMutilations génitales féminines/excision – a) Femmes : pour
None/tNone/tJustification de la violence conjugale – Pourcentage de fem
None/tNone/tDiscipline imposée par la violence – Pourcentage d’enfants
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tSOURCES PRINCIPALES DES DONNÉES/tNone/tNone/tNone/tNone/tNo
None/tNone/tTravail des enfants – Enquêtes en grappes à indicateurs mul
None/tNone/tMariage d’enfants – MICS, EDS et autres enquêtes nationales
None/tNone/tEnregistrement des naissances – MICS, EDS, autres enquêtes
None/tNone/tMutilations génitales féminines/excision – MICS, EDS et aut
None/tNone/tJustification de la violence conjugale – MICS, EDS et autre
None/tNone/tDiscipline imposée par la violence – MICS, EDS et autres en
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tDEFINICIONES DE LOS INDICADORES/tNone/tNone/tNone/tNo
None/tNone/tNone/tTrabajo infantil – Porcentaje de niños de 5 a 14 años
None/tNone/tNone/tMatrimonio precoz – Porcentaje de mujeres de 20 a 24
None/tNone/tNone/tInscripción del nacimiento – Porcentaje de niños y ni
None/tNone/tNone/tMutilación/excisión genital femenina – (a) Mujeres –
None/tNone/tNone/tJustificación de golpear a la mujer – Porcentaje de m
None/tNone/tNone/tDisciplina violenta – Porcentaje de niños de 2 a 14 a
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
```

```
None/tNone/tNone/tFUENTES ESTADÍSTICAS PRINCIPALES/tNone/tNone/tNone/tN
None/tNone/tNone/tTrabajo infantil – Encuestas de Indicadores Múltiples
None/tNone/tNone/tMatrimonio precoz – MICS, DHS y otras encuestas nacio
None/tNone/tNone/tInscripción del nacimiento – MICS, DHS, otras encuest
None/tNone/tNone/tMutilación/excisión genital de la mujer – MICS, DHS y
None/tNone/tNone/tJustificación de golpear a la mujer – MICS, DHS y otr
None/tNone/tNone/tDisciplina violenta – MICS, DHS y otras encuestas nac
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
None/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/tNone/
```

```python
# Print the contents of each row and cells, also improve readability
# iterate over each row
for row_index, row_values in enumerate(sheet.iter_rows(min_row=1, values_or
  row_name = f"Row {row_index}"
  print(row_name)
  # iterate through each cell in the row
  for cell_index, cell_value in enumerate(row_values, start=1):
    print(f"  Cell {cell_index}: {cell_value}")
  # improve readability by adding a separator between each row
  print("-" * 20)
```

```
    Cell 9: None
    Cell 10: None
    Cell 11: None
    Cell 12: None
    Cell 13: None
    Cell 14: None
    Cell 15: None
    Cell 16: None
    Cell 17: None
    Cell 18: None
    Cell 19: None
    Cell 20: None
    Cell 21: None
    Cell 22: None
    Cell 23: None
    Cell 24: None
    Cell 25: None
    Cell 26: None
```

```
  Cell 27: None
  Cell 28: None
  Cell 29: None
  Cell 30: None
  Cell 31: None
  Cell 32: None
  Cell 33: None
  Cell 34: None
  Cell 35: None
  Cell 36: None
  Cell 37: None
  Cell 38: None
  Cell 39: None
  Cell 40: None
  Cell 41: None
  Cell 42: None
  Cell 43: None
  Cell 44: None
  Cell 45: None
  Cell 46: None
  Cell 47: None
  Cell 48: None
  Cell 49: None
--------------------
Row 349
  Cell 1: None
  Cell 2: None
  Cell 3: None
  Cell 4: None
  Cell 5: None
  Cell 6: None
  Cell 7: None
  Cell 8: None
  Cell 9: None
  Cell 10: None
  Cell 11: None
  Cell 12: None
  Cell 13: None
  Cell 14: None
  Cell 15: None
  Cell 16: None
  Cell 17: None
```

```
# skip to the header string "Countries and areas"
start_row = None
# iterate over the data
for row_index, row_values in enumerate(sheet.iter_rows(min_row=1, values_or
  # check if the row contains the header string
  if "Countries and areas" in row_values:
    # if found, go to the next row
    start_row = row_index + 1
    break
# dictionary to store extracted data
extracted_data = {}
```