In [7]:
```
#https://docs.google.com/spreadsheets/d/1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc/edit#gid=0
#https://docs.google.com/spreadsheets/d/1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc/edit?usp=sharing

import pandas as pd

sheet_id ='1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc'
xls=pd.ExcelFile(f"https://docs.google.com/spreadsheets/d/{sheet_id}/export?format=xlsx")
year_2021 = pd.read_excel(xls,'2021', header=1)
year_2021
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\io\parsers\python_parser.py:709, in PythonParser._next_line(self)
    708 try:
--> 709     line = self._check_comments([self.data[self.pos]])[0]
    710     self.pos += 1

IndexError: list index out of range

During handling of the above exception, another exception occurred:

StopIteration                             Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\io\parsers\python_parser.py:400, in PythonParser._infer_columns(sel
f)
    399     while self.line_pos <= hr:
--> 400         line = self._next_line()
    402 except StopIteration as err:

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\python_parser.py:722, in PythonParser._next_line(self)
    721         except IndexError:
--> 722             raise StopIteration
    723 else:

StopIteration:

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
Cell In[7], line 8
      6 sheet_id ='1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc'
      7 xls=pd.ExcelFile(f"https://docs.google.com/spreadsheets/d/{sheet_id}/export?format=xlsx")
----> 8 year_2021 = pd.read_excel(xls, '2021', header=1)
      9 year_2021

File ~\anaconda3\Lib\site-packages\pandas\io\excel\_base.py:486, in read_excel(io, sheet_name, header, names,
index_col, usecols, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_de
fault_na, na_filter, verbose, parse_dates, date_parser, date_format, thousands, decimal, comment, skipfooter,
storage_options, dtype_backend)
    480     raise ValueError(
    481         "Engine should not be specified when passing "
    482         "an ExcelFile - ExcelFile already has the engine set"
    483     )
    485 try:
--> 486     data = io.parse(
    487         sheet_name=sheet_name,
    488         header=header,
    489         names=names,
    490         index_col=index_col,
    491         usecols=usecols,
    492         dtype=dtype,
    493         converters=converters,
    494         true_values=true_values,
    495         false_values=false_values,
    496         skiprows=skiprows,
    497         nrows=nrows,
    498         na_values=na_values,
    499         keep_default_na=keep_default_na,
    500         na_filter=na_filter,
    501         verbose=verbose,
    502         parse_dates=parse_dates,
    503         date_parser=date_parser,
    504         date_format=date_format,
    505         thousands=thousands,
    506         decimal=decimal,
    507         comment=comment,
    508         skipfooter=skipfooter,
    509         dtype_backend=dtype_backend,
    510     )
    511 finally:
    512     # make sure to close opened file handles
    513     if should_close:

File ~\anaconda3\Lib\site-packages\pandas\io\excel\_base.py:1551, in ExcelFile.parse(self, sheet_name, heade
r, names, index_col, usecols, converters, true_values, false_values, skiprows, nrows, na_values, parse_dates,
date_parser, date_format, thousands, comment, skipfooter, dtype_backend, **kwds)
   1518 def parse(
   1519     self,
   1520     sheet_name: str | int | list[int] | list[str] | None = 0,
   (...)
   1538     **kwds,
   1539 ) -> DataFrame | dict[str, DataFrame] | dict[int, DataFrame]:
   1540     """
   1541     Parse specified sheet(s) into a DataFrame.
   1542 
   (...)
   1549         DataFrame from the passed in Excel file.
   1550     """
-> 1551     return self._reader.parse(
   1552         sheet_name=sheet_name,
   1553         header=header,
   1554         names=names,
   1555         index_col=index_col,
   1556         usecols=usecols,
   1557         converters=converters,
   1558         true_values=true_values,
   1559         false_values=false_values,
   1560         skiprows=skiprows,
   1561         nrows=nrows,
   1562         na_values=na_values,
   1563         parse_dates=parse_dates,
   1564         date_parser=date_parser,
   1565         date_format=date_format,
   1566         thousands=thousands,
   1567         comment=comment,
   1568         skipfooter=skipfooter,
   1569         dtype_backend=dtype_backend,
   1570         **kwds,
   1571     )

File ~\anaconda3\Lib\site-packages\pandas\io\excel\_base.py:889, in BaseExcelReader.parse(self, sheet_name, h
eader, names, index_col, usecols, dtype, true_values, false_values, skiprows, nrows, na_values, verbose, pars
e_dates, date_parser, date_format, thousands, decimal, comment, skipfooter, dtype_backend, **kwds)
    887     except Exception as err:
    888         err.args = (f"{err.args[0]} (sheet: {asheetname})", *err.args[1:])
--> 889         raise err
    891 if last_sheetname is None:
    892     raise ValueError("Sheet name is an empty list")

File ~\anaconda3\Lib\site-packages\pandas\io\excel\_base.py:851, in BaseExcelReader.parse(self, sheet_name, h
eader, names, index_col, usecols, dtype, true_values, false_values, skiprows, nrows, na_values, verbose, pars
e_dates, date_parser, date_format, thousands, decimal, comment, skipfooter, dtype_backend, **kwds)
    849 # GH 12292 : error when read one empty column from excel file
    850 try:
--> 851     parser = TextParser(
    852         data,
    853         names=names,
    854         header=header,
    855         index_col=index_col,
    856         has_index_names=has_index_names,
    857         dtype=dtype,
    858         true_values=true_values,
    859         false_values=false_values,
    860         skiprows=skiprows,
    861         nrows=nrows,
    862         na_values=na_values,
    863         skip_blank_lines=False,  # GH 39808
    864         parse_dates=parse_dates,
    865         date_parser=date_parser,
    866         date_format=date_format,
    867         thousands=thousands,
    868         decimal=decimal,
    869         comment=comment,
    870         skipfooter=skipfooter,
    871         usecols=usecols,
    872         dtype_backend=dtype_backend,
    873         **kwds,
    874     )
    876     output[asheetname] = parser.read(nrows=nrows)
    878     if header_names:

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1803, in TextParser(*args, **kwds)
   1748 """
   1749 Converts lists of lists/tuples into DataFrames with proper type inference
   1750 and optional (e.g. string to datetime) conversion. Also enables iterating
   (...)
   1800     .. versionchanged:: 1.2
   1801 """
   1802 kwds["engine"] = "python"
-> 1803 return TextFileReader(*args, **kwds)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1407, in TextFileReader.__init__(self, f, eng
ine, **kwds)
   1404     self.options["has_index_names"] = kwds["has_index_names"]
   1406 self.handles: IOHandles | None = None
-> 1407 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\readers.py:1679, in TextFileReader._make_engine(self, f,
engine)
   1676     raise ValueError(msg)
   1678 try:
-> 1679     return mapping[engine](f, **self.options)
   1680 except Exception:
   1681     if self.handles is not None:

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\python_parser.py:124, in PythonParser.__init__(self, f,
**kwds)
    118 self._col_indices: list[int] | None = None
    119 columns: list[list[Scalar | None]]
    120 (
    121     columns,
    122     self.num_original_columns,
    123     self.unnamed_cols,
--> 124 ) = self._infer_columns()
    126 # Now self.columns has the set of columns that we will process.
    127 # The original set is stored in self.original_columns.
    128 # error: Cannot determine type of 'index_names'
    129 (
    130     self.columns,
    131     self.index_names,
   (...)
    136     self.index_names,  # type: ignore[has-type]
    137 )

File ~\anaconda3\Lib\site-packages\pandas\io\parsers\python_parser.py:410, in PythonParser._infer_columns(sel
f)
    408     joi = list(map(str, header[:-1] if have_mi_columns else header))
    409     msg = f"[{','.join(joi)}], len of {len(joi)}, "
--> 410     raise ValueError(
    411         f"Passed header={msg}"
    412         f"but only {self.line_pos} lines in file"
    413     ) from err
    415 # We have an empty file, so check
    416 # if columns are provided. That will
    417 # serve as the 'line' for parsing
    418 if have_mi_columns and hr > 0:

ValueError: Passed header=[1], len of 1, but only 1 lines in file (sheet: 2021)
```

In [8]:
```
#https://docs.google.com/spreadsheets/d/1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc/edit#gid=0
#https://docs.google.com/spreadsheets/d/1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc/edit?usp=sharing

import pandas as pd
```

In [9]:
```
sheet_id ='1hvgztPedGtNFV6fjjKhTts_UKMs0rsCF5YIJbMnNLTc'
```

In [11]:
```
xls = pd.ExcelFile(f"https://docs.google.com/spreadsheets/d/{sheet_id}/export?format=xlsx")
```

In [13]:
```
year_2021 = pd.read_excel(xls,'2021')
```

In [14]:
```
year_2021
```

Out[14]:
```
  5  6
```

In [ ]: