```
import pandas as pd
from extractors.direct downloads import extract with request
from models.registry import Registry
from utils.helpers import clean str, date to iso, date from pdf
class CYM(Registry):
   def __init__(self, **kwargs):
       Child container for extracting, transforming, and loading registry data for the Cayman Islands
        :param kwargs: see models.registry for details
       # assign constant values for final registry
        and a second and a female
        mot in attitude of from
       mod article, name it "Opini broatons betterring of the fugues belonds"
        mod responsible roles in from
        # assign registry input data source and filename
       mich amprecia, ser in Nersye chama assessment and the
        mid registry friends i factors borough beginning per
        # assign registry input data extraction method
        # assign updated-date scraping method for registry input data
        # assign registry input data reading method
        # assign registry data wrangling method
        # extract, transform, and load registry data
   def extract registry(self):
       Extracts registry input data from remote source
        :return: effect - creates [DATA DIR|TEST DIR]/[REGISTRY INPUT DIR]/[self.country code]/[self.registry filename] file
        # extract registry input data
        regionary is extraord, an
                                       med regions, art, and regions, friends, regions, referred regions, reti-
        # scrape updated-date for registry input data
   def scrape_registry_updated(self, registry=None):
        11 11 11
       Scrapes and formats updated-date for registry input data from remote source
       :param (io.BytesIO) registry: bytes stream of extracted registry input data
        :return: effect - modifies self.updated
              ___
            # handle when registry input data has been extracted but not in current session
       date it date, from patrosportus, affabilit nacionalizati, from sensiti
        # assign updated-date for registry input data
          def read registry(self):
       Reads registry input data
        :return: effect - modifies self.registry
       # read registry input data
        # prep registry input data
        ____
        regional recipies "I (sect.", formal referencies) for set in regional recipies.
        # assign registry input data
   def wrangle registry(self):
       Transforms registry data
        :return: effect - modifies self.wrangled records
        # loop through raw registrations
            # initialize storage for wrangled representation of registration in focus
                "Strant, Anne": Ann. In. Section. Sens. of Depositionism, "Sections/"s.
                "specif": you despond on our
                "barrot's francis": 100 Barrots
                "model" on Both
                "section, so": see Section, Robbins,
                "case": con desportanted, famous
                "ca, meser": form
            sect emergine, records appendicement
         _format_registry(self, registry):
       Cleans and merges tables across multiple pdf pages
        :param (camelot.core.TableList) registry: tables of registrations from registry pdf to process
        :return: cleaned and merged transformation of 'registry' argument
        # initialize storage for tables on multiple pages
        # initialize storage for column names
          ....
        # loop through pdf pages
            # assign table on the page in focus
            ....
            ---
                # assign column names once using the table on the first page
            # drop header row from each table
           # merge tables across multiple pages together
        regionary and make a breaker
   @staticmethod
   def \underset{"""}{\text{extract}}_colnames(header, colsize):
       Parses and validates table column names that are incorrectly captured by the camelot extraction process
        :param (str) header: table column names from camelot reading method to extract values from
        :param (int) colsize: expected number of columns the header values should conform to
        :return: formatted copy of 'header' argument to label table columns
       breaker it breaker against a "or"s
               beaders I accords and beaders in the Nacot's
            # handle when multi-line column name is treated as two distinct labels
            .....
       ---
   @staticmethod
   def __format_colname(colname):
       Formats column name for namedtuple compatibility
        :param (str) colname: column name from registry input data to format
        :return: formatted copy of 'colname' argument suitable for namedtuple indexing
```

mineral in class, and recipied angles of the 1,75

import camelot