Get_gage_no.py:
Get the data from the excel file: copy of Illinois gages 2022-3225.xlsx
Then, convert the datatype of site_no into a string list.
Since the site number in the URL is 8 digits, I add the zero before the site number from the excel file.

Gage_to_excel.py:

A. Designing flow class:
   1. def __init__: call the river_number
   2. def scrape(self): collect the daily data of river_number from the website
   3. def excel_list(self): get the data frames (
      new_df: from the first date to the end
      Ym_df: the data in a month/year
      Y_df: the data in a year
      After_df: the data frame of filtering over missing 10 days of a year
      Final_df: calculate Q7.10 for each year) and export an excel file)
B. Scraping data from the website:
   1. Use the requests.get function to obtain data from the USGS.
   2. The today variable helps us update the URL.
   3. After getting the data, I use lthe xml parser to parse them, and the prettify function helps export them as HTML format.
   4. Use regular expression to collect the site number, year, month, date, and flow.
C. New data frame
   1. df is the dates' data frame with values
   2. new_df is the data frame with all dates from start to nowadays and including zero values.
   3. The pd.date_range function helps collect all dates of each site.
D. Year_month data frame
   1. Use the lambda to extract only Year and Month from df['Date']
   2. Use the dictionary to count the number of collected days of each month from df, then build Ym_df
   3. The all_dates dataframe with all date values combines the Ym_df.
   4. Since i got the number of collected dates, I define the days_in_month function to provide the number of days in a month.
E. Year data frame
   1. Use the dictionary to count collected dates and build as Y_df.
   2. Pair Y_df with the all_dates data frame.
   3. Use the calendar.isleap to obtain the number of days in a year.

4. Then count the differences between collected_days and days_in_year.

5. The count which is lower than ten, the year data would be used as final data.

6. Use a dictionary to find the Minimum_of_flow7d, row by row

F. After data frame

1. Select those years whose missing days are lower than 10 days from new_df.

2. Calculate the percentile of each flow

G. Calculation

1. Copy rows that are True in the Can_use_data column from the Y_df to the Final_df.

2. Calculate log with Minimum_of_flow7d.

3. k is the skew of a log with Minimum_of_flow7d.

H. Convert data frames into excel sheets

Percentile.py

A. Designing flow class:

1. def __init__: call the river_number

2. def scrape(self): collect the daily data of river_number from the website

3. def excel_list(self): get the data frames (

new_df: from the first date to the end

B. Calculate percentile of each day

1. Using flow column in the new_df data frame to calculate the percentile