```python
#!/usr/bin/env python
import logging
import numpy
import csv
from config import *
import pyclowder.extractors as extractors

def main():
    global extractorName, messageType, rabbitmqExchange, rabbitmqURL, logger, registrationEndpoints

    # Set logging
    logging.basicConfig(format='%(asctime)-15s %(levelname)-7s : %(name)s - %(message)s', level=logging.WARN)
    logging.getLogger('pyclowder.extractors').setLevel(logging.INFO)
    logger = logging.getLogger('extractor')
    logger.setLevel(logging.DEBUG)

    extractors.setup(extractorName=extractorName, messageType=messageType, rabbitmqURL=rabbitmqURL,
                     rabbitmqExchange=rabbitmqExchange)

    # Register extractor
    extractors.register_extractor(registrationEndpoints)

    # Connect to extractor and wait for files
    extractors.connect_message_bus(extractorName=extractorName, messageType=messageType,
                                   rabbitmqURL=rabbitmqURL,
                                   rabbitmqExchange=rabbitmqExchange,
                                   processFileFunction=process_file,
                                   checkMessageFunction=None)

def process_file(parameters):
    global field_name

    inputfile = parameters['inputfile']

    mean_desc = 'Mean total monthly precipitation'
    median_desc = 'Median total monthly precipitation'
    standard_dev_desc = 'Standard deviation of total monthly precipitation'

    # Initialize a records list and field name
    records = list()
    field_name = 'TPCP'

    # Read csv file contents into a variable
    csv_file = open(inputfile, 'rb')
    data_table = csv.DictReader(csv_file)

    # Read each row of the table and append the data into records array
    for data_row in data_table:
        records.append(float(data_row[field_name]))

    # Calculate mean, median and standard deviation and upload it back as metadata to Clowder
    mean = numpy.mean(records)
    median = numpy.median(records)
    std = numpy.std(records)

    # Context url
    context_url = "https://clowder.ncsa.illinois.edu/contexts/metadata.jsonld"

    # Store results as metadata
    metadata = {
        "@context": [context_url,
```

```
60                          {
61                              "mean": "http://clowder.ncsa.illinois.edu/"+ extractorName +
                                "#mean",
62                              "median": "http://clowder.ncsa.illinois.edu/" + extractorName
                                + "#median",
63                              "std": "http://clowder.ncsa.illinois.edu/" + extractorName +
                                "#std",
64                              "value": "http://clowder.ncsa.illinois.edu/" + extractorName +
                                "#value",
65                              "description": "http://clowder.ncsa.illinois.edu/"+
                                extractorName + "#description"
66                          }],
67          "attachedTo": {"resourceType": "file", "id": parameters["fileid"]},
68          "agent": {
69              "@type": "cat:extractor",
70              "extractor_id":
                "https://clowder.ncsa.illinois.edu/clowder/api/extractors/"+ extractorName
71          },
72          "content": {
73              "mean": {
74                  "value": round(mean, 3),
75                  "description": mean_desc
76              },
77              "median": {
78                  "value": round(median, 3),
79                  "description": median_desc
80              },
81              "std": {
82                  "value": round(std, 3),
83                  "description": standard_dev_desc
84              }
85          }
86      }
87
88      # upload metadata
89      extractors.upload_file_metadata_jsonld(mdata=metadata, parameters=parameters)
90      logger.info("Uploaded metadata %s", metadata)
91
92
93  if __name__ == '__main__':
94      main()
95
```