

농업연구자의 기상자료 활용을 위한 파이썬 패키지 제작

양현지¹, 박주현^{1*}, 안문일¹, 강민구², 한용규¹, 박은우¹

¹에피넷, ²국립농업과학원

(2022년 11월 03일 접수; 2023년 01월 13일 수정; 2023년 04월 05일 수락)

Python Package Production for Agricultural Researcher to Use Meteorological Data

Hyeon Ji Yang¹, Joo Hyun Park^{1*}, Mun-Il Ahn¹, Min Gu Kang²,
Yong Kyu Han¹, Eun Woo Park¹

¹EPINET Co., Ltd., Anyang 14056, Korea

²National Institute of Agricultural Sciences, RDA, Wanju 55365, Korea

(Received November 03, 2022; Revised January 13, 2023; Accepted April 05, 2023)

ABSTRACT

Recently, the abnormal weather events and crop damages occurred frequently likely due to climate change. The importance of meteorological data in agricultural research is increasing. Researchers can download weather observation data by accessing the websites provided by the KMA (Korea Meteorological Administration) and the RDA (Rural Development Administration). However, there is a disadvantage that multiple inquiry work is required when a large amount of meteorological data needs to be received. It is inefficient for each researcher to store and manage the data needed for research on an independent local computer in order to avoid this work. In addition, even if all the data were downloaded, additional work is required to find and open several files for research. In this study, data collected by the KMA and RDA were uploaded to GitHub, a remote storage service, and a package was created that allows easy access to weather data using Python. Through this, we propose a method to increase the accessibility and usability of meteorological data for agricultural personnel by adopting a method that allows anyone to take data without an additional authentication process.

Key words: Agricultural researcher, Weather data, GitHub, Python package, Remote storage



* Corresponding Author : Joo Hyun Park
(parkjh@epinet.kr)

I. 배경 및 요약

농업은 기상에 매우 민감한 산업이다. 농업 분야에서의 기후변화는 농작물의 재배 적지를 변화시키고, 생산량이나 품질에 영향을 줄 뿐 아니라, 빈발한 이상 기상의 경우 농업 생산성 저하의 원인이 된다(Park *et al.*, 2021). 이상기상 현상이란 기온, 강수량 등의 기후 요소가 평년값에 비해 크게 벗어난 기상현상을 의미하며, 세계기상기구(WMO; World Meteorological Organization)에서는 25년에 1회 정도밖에 나타나지 않는 특이하고 아주 드문 기상현상으로 정의한다(Sim *et al.*, 2018). 2020년 이상기후 보고서(KMA, 2020)에 의하면, 농작물의 개화기인 4월에 이상 저온으로 인해 과수 및 발작물 등 43,554ha에 해당하는 농작물에 피해가 발생했다. 7~8월에는 기록적으로 긴 장마와 많은 강수량으로 인해 33,492ha의 농작물에 침수 피해가 발생했다. 2021년에도 1월 한파, 4월 이상 저온, 7월 집중호우 및 7~8월 폭염 등 이상기상에 의한 농작물 피해가 계속되었다(KMA, 2021). 이렇게 기후변화로 인한 이상기상 빈도가 매년 증가할 경우, 향후 우리나라 식량안보에 큰 위협 요소로 작용할 수 있다(Jeong *et al.*, 2020). 따라서 농업 분야의 기상을 이용한 연구는 더욱 중요해지고 있다.

기상청에서는 기상자료개방포털(<https://data.kma.go.kr>)을 통해 지상, 해양, 고층, 항공관측, 위성, 레이더, 수치예보모델자료 등 총 30종류의 날씨 데이터를 제공하고 있다. 기상청에서 제공하는 중관기상관측(ASOS, Automated Synoptic System)지점은 현재 96개, 자동기상관측(AWS, Automatic Weather System) 지점은 527개, 농업기상관측(AAOS, Automated Agriculture Observing System) 지점은 11개 지점이다. 농촌진흥청에서는 농업기상정보 서비스인 농업날씨365(<https://weather.rda.go.kr>)를 통해 약 211개 지점의 자동기상관측 자료를 제공하고 있다. 두 사이트에 접속하면 누구나 연구에 필요한 기상관측자료를 받을 수 있지만, 대량의 기상자료가 필요할 때는 여러 번의 설정이 필요한 단점이 있다. 예를 들어, 기상자료개방포털은 로그인 한 회원을 기준으로 일별 자료는 최대 10년까지 시간자료는 1년, 분자료는 1일 자료만 조회 및 다운이 가능하고 그 외에는 지점 및 년도 별로 묶인 파일셋을 다운 받아야 한다. 농업날씨365 역시 일반 연구자가 사이트에 접속해 기상자료를 받고자 하는 경우, 한 번에 한 지점씩 1년 미만의 기간에 대해 조회할 수 있다.

이마저도 기간으로 조회할 경우 습도, 풍향, 풍속, 일조 시간, 토양수분 등 농업 연구에 필요한 정보는 제공되지 않아 일별로 따로 조회해야 하는 번거로움이 존재한다. 이렇게 자료를 모두 다운로드 했다고 하더라도 여러 개의 파일을 컴퓨터에 저장한 후 작업을 위해 다시 열어야 하는 번거로운 작업이 한번 더 수반된다. 또, 연구자는 간단한 조회작업을 위해서도 매번 파일을 다운로드해 확인해야 한다.

매번 이 번거로운 작업을 피하고자 각 연구자가 연구에 필요한 자료를 독립적인 로컬 컴퓨터에 저장하고 관리하는 것에는 비효율적인 면이 있다. 예를 들어 의도치 않은 수정이나 손상을 감지할 방법이 없고 요즘과 같이 노트북, 데스크톱, 재택근무로 인한 멀티 디바이스가 일반화된 환경에서는 자료 복제의 불편함이 수반된다. 또한 관리하는 데이터가 늘수록 원하는 데이터를 찾거나 가진 데이터를 관리하는 시간도 그만큼 증가하게 된다. 네트워크의 발달로 빠른 데이터 통신이 가능해진 요즘, 데이터 사용자들이 언제 어디서나 다양한 디바이스를 통해 자신의 데이터에 접근할 수 있도록 돕는 각종 원격 저장소 서비스가 등장하고 있다(Lee and Lee, 2011). Cho *et al.*(2020)은 전남 보성의 영농형 태양광 하부에서 가을보리 재배 기간 동안 관측된 미기상 자료를 원격 저장소 호스팅 서비스 중 하나인 깃허브(GitHub)에 업로드하여 자료를 공유하는 데 이용했다. Kim *et al.*(2021)은 또한 전남농업기술원 실험 논 포장에서 관측된 직달일사량과 산란일사량을 csv 파일로 정리하여 깃허브에 업로드 함으로서 공개하였다.

깃허브(GitHub)는 분산 버전관리 시스템인 깃(Git)을 기반으로 하는 오픈 소스 소셜 코딩 서비스로, 소프트웨어 저장과 버전 관리, 개발자 커뮤니티의 협업과 온라인 교류를 위한 다양한 기능을 제공한다. 깃허브의 핵심 기능은 깃의 원격 저장소 호스팅인데, 이를 통해 프로그램 소스뿐 아니라 일반 파일 및 자료를 쉽게 공유하고 관리할 수 있는 환경을 제공한다(Kang *et al.*, 2017). 2022년 10월, 깃허브에서는 프로그래밍 언어 인기도(PYPL, Popularity of Programming Language)를 통해 파이썬(Python)이 지난 5년간 가장 많이 성장했으며, 현재 전 세계적으로 가장 인기 있는 언어라고 발표했다.

파이썬이 웹이나 데이터베이스 프로그래밍뿐만 아니라 다양한 분야에서 관심을 받는 이유는 수치 연산의 프로그래밍 및 데이터 분석, 사물인터넷(IoT) 분야

등 다양한 공학 분야에서 활용성과 효율성을 가지기 때문이다. 기본적으로 파이썬은 오픈소스로, 누구나 무료로 다운로드 해 사용할 수 있으며 데이터 통계 분석 및 시각화에 용이한 Numpy, Scipy, matplotlib등을 포함해 다양한 분야의 방대한 라이브러리를 제공한다(Lee, 2016). 농업연구에서도 파이썬은 각종 작물 및 병해충 모형 개발이나 분석을 위해 자주 사용되는 언어이다. Jang et al. (2020) 등은 국내 공공기상데이터를 이용해 전날의 토양수분 및 기상정보를 가지고 다음 날의 평균 토양수분을 예측하는 5가지 모델을 파이썬 언어를 이용해 제작하고 성능을 비교했다. Hur et al. (2020) 은 남한 지역 기후 변화량 평가를 위해 생산한 고해상도 농업기후 자료 정보를 파이썬 패키지를 이용해 공개한 바 있다.

기상청과 농촌진흥청에서 대량의 기상자료를 받는 방법으로는 기상자료개방포털과 공공데이터포털(https://www.data.go.kr)에서 제공하는 오픈 API를 이용하는 방법도 존재한다. API (Application Programming Interface)란 서로 다른 프로그램의 기능이나 데이터를 상호 이용할 수 있도록 미리 정한 통신 규칙이며, 외부 제3자도 접근할 수 있도록 표준화하여 공개한 것을 오픈 API라 한다(Noh and Son, 2021). 공공데이터포털에서도 각 공공기관이 보유한 공공데이터를 오픈 API를 통해 무료로 제공하고 있어 사용자는 ‘사용요청’ 후 발급받은 오픈 API 인증키를 이용해 데이터를 받을 수 있다. API는 자료 및 서비스를 표준화하여 제공하기 때문에 누구나 동일한 데이터를 받을 수 있다. 그러나 오픈 API를 통한 자료는 제공하는 메소드 형태가 제한적이며, 사용자는 JSON 형태의 파일을 읽기 위해 환경을 구성하고 원하는 자료를 원하는 형태로 가져오기 위한 추가 작업이 필요하다. API 인증키는 개인에게만 할당되는 것으로, 사용자마다 인증키를 발급받아야 한다는 번거로움도 존재한다.

본 연구에서는 원격 저장소 서비스에 자료를 업로드하고 API 키 확인 등과 같은 추가적인 인증 절차 없이 누구나 자료를 가져갈 수 있는 방식을 채택하여 농업 관계자들의 기상자료에 대한 접근성 및 활용성을 높이는 방법을 제안한다. 이를 위해 기상자료를 수집한 후 앞서 설명한 깃허브에 업로드 하고 소프트웨어 프로그램인 파이썬을 이용해 기상자료에 쉽게 접근할 수 있는 패키지를 제작한 후 이를 평가하고자 한다.

II. 자료 및 방법

2.1. 기상자료 수집 및 처리

기상청 방재기상관측기(AWS)와 종관기상관측기(ASOS)의 일별 관측(또는 집계) 자료와 농촌진흥청의 기상관측기(AWS)의 일별 농업기상 관측자료를 수집하였다. 기상청 기상자료는 기상자료개방포털(https://data.kma.go.kr)의 ‘종관기상관측’, ‘방재기상관측’을 이용했으며, 종관기상관측자료는 1971년부터 2021년까지, 방재기상관측자료는 1996년 12월 31일부터 2021년까지 수집하였다. 농촌진흥청 기상자료는 공공데이터포털(https://www.data.go.kr)에서 제공하는 농촌진흥청 농업기상관측자료 API를 이용해 2010년부터 2022년 8월 30일까지 수집하였다. 이렇게 수집한 기상청 자료는 결측과 누락을 포함하여 물리 한계 검사, 기후 범위 검사, 지속성 검사를 수행하여 이상 값을 탐지하였다. 기상청 데이터 품질관리 규정 제 5조에 따라 종관기상관측자료와 방재기상관측자료의 데이터 품질 담당부서는 기상청 국가기후데이터센터이다. 따라서 이상 값 탐지는 국가기후데이터센터에서 제공하는 “기상관측데이터 품질검사 프로세스”를 준

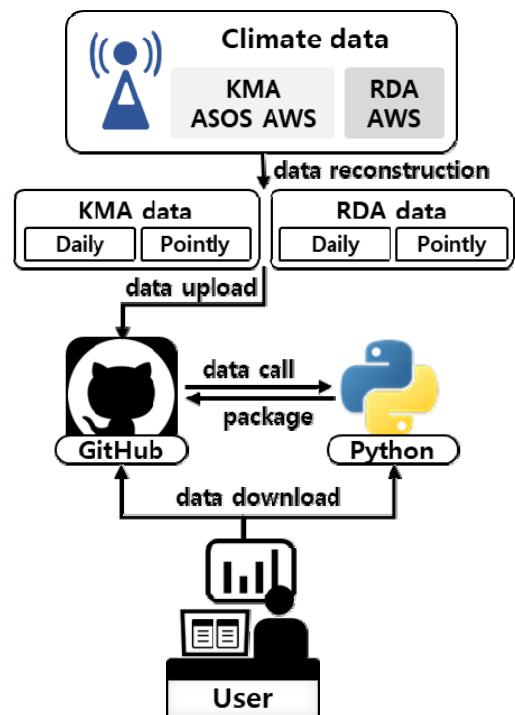


Fig. 1. Meteorological data redistribution flow chart.

용하였다. 농촌진흥청의 경우 별도의 규정이 존재하지 않으므로 결측과 누락에 대한 이상 탐지를 확인하였다. 그 후 NCAM (National Center for Agro Meteorology) 에서 제공한 기상청의 기상관측 raw 자료와 국립농업과학원에서 운영하는 농업날씨365(<https://weather.rda.go.kr>)를 참고해 탐지된 이상 자료의 복구를 시도하였으며, 해당 자료로 보완할 수 없는 부분은 추가적인 작업을 진행하지 않았다.

위 방법으로 처리한 기상청과 농촌진흥청 기상관측

자료는 일별, 지점별로 재구성한 후 각각 깃허브 저장소에 오픈 프로젝트로 공유해 누구나 쉽게 자료에 접근할 수 있게 하였다(<https://github.com/dogbull/kawos>, <https://github.com/yanghjep/rdawos>).

2.2. 패키지 제작

깃허브를 참조하여 업로드된 기상자료를 제공할 수 있는 방법으로 Java, C, C++, C#, 파이썬, JavaScript, Visual Basic, NET, R, PHP 등 수많은 프로그래밍

```

import datetime
import os

import pandas as pd
import geopandas as gpd
from dateutil.parser import parse as str2datm

from .mapper import STATION_LIST_HEADER_MAPPER

BASE_URL = 'https://raw.githubusercontent.com/dogbull/kawos/master/'

def read(sub_url, cache_dir):
    if cache_dir:
        cache_path = f'{cache_dir}/{sub_url}'
    else:
        cache_path = None

    if cache_path and os.path.exists(cache_path):
        df = pd.read_csv(cache_path)
    else:
        data_url = f'{BASE_URL}/{sub_url}'
        df = pd.read_csv(data_url)
        if cache_path:
            os.makedirs(os.path.dirname(cache_path), exist_ok=True)
            df.to_csv(cache_path, index=False)
    col_set = set(df.columns)
    if len({'lon', 'lat'} - col_set) == 0:
        geometry = gpd.points_from_xy(x=df.lon, y=df.lat)

```

```

def read_single_point(code, begin, until, *, cache_dir='./rdawos'):
    if isinstance(begin, (datetime.date,)):
        begin_year = begin.year
    elif isinstance(begin, (str,)):
        begin = str2datm(begin).date()
        begin_year = begin.year
    else:
        begin_year = int(begin)

    if isinstance(until, (datetime.date,)):
        until_year = until.year
    elif isinstance(until, (str,)):
        until = str2datm(until).date()
        until_year = until.year
    else:
        until_year = int(until)

    df_ary = []
    for year in range(begin_year, until_year + 1):
        df = read(f'pointly/{code}/{year}/{code}.{year}.csv', cache_dir)
        df_ary.append(df)
    df = pd.concat(df_ary)

    if isinstance(begin, (datetime.date,)):
        df = df[df['tm'] >= begin.strftime('%Y-%m-%d')]
    if isinstance(until, (datetime.date,)):
        df = df[df['tm'] <= until.strftime('%Y-%m-%d')]

    return df.reset_index()

```

Fig. 2. A python code snippet of 'pykawos'(L) and 'pyrdawos'(R).

Table 1. Functions provided by 'pykawos' and 'pyrdawos'

Package	Function	Description
pykawos	read_stations	Inquiry asos/aws station list ex) read_asos_stations() / read_stations('asos') or read_aws_stations() / read_stations('aws')
	read_single_point	Inquire data from a single asos/aws point ex) read_single_point('asos', 108, '20210101', '20211231') or read_asos_single_point(108, '20210101', '20211231')
	read_multi_point	Inquire data from a asos/aws multi point ex) read_multi_point('asos', '20210101') or read_aws_multi_point('20210101')
pyrdawos	read_stations	Inquiry all station list ex) read_stations
	read_single_point	Inquire data from a single point ex) read_single_point('13718A001', '20210101', '20211231')
	read_multi_point	Inquire data from a multi point ex) read_multi_point('20210101')

언어들이 존재한다. 본 연구에서는 그 중 파이썬을 이용했으며, 파이썬에서 활용할 수 있는 pandas, geopandas, dateutil 모듈을 이용해 깃허브에 업로드되어있는 기상자료를 불러올 수 있게 하는 패키지인 ‘pykawos’, ‘pyrdawos’를 제작하였다.

두 패키지는 각각 기상청의 종관기상과 방재기상, 농촌진흥청의 기상자료를 불러올 수 있으며, 각 기상자료의 관측지점 역시 프로그램 함수 호출 한 번으로 간단하게 조회할 수 있다. 기상자료 조회 시에는 특정 지점에 대한 장기간 자료를 다운받을 수 있거나 모든 지점의 특정 날짜의 자료를 다운받을 수 있게 하였다. 해당 패키지는 PyPI (The Python Package Index; <https://pypi.org>)에 등록되어 있어 pip 명령어를 통해 간단히 설치할 수 있다. 또한 깃허브에 프로젝트를 업로드(<https://github.com/dogbull/pykawos>, <https://github.com/yanghjep/pyrdawos>) 해 사용법과 소스 코드를 확인할 수 있게 하였다.

III. 사용 방법

개발한 패키지 중 ‘pykawos’의 간단한 응용사례를 보이기 위해 구글 코랩(Google Colab)을 이용했다. 사용자가 작성한 파이썬 프로그램을 실행하기 위해서는 파이썬 인터프리터를 다운 받고 프로그램 실행에 필요한 외부 라이브러리를 설치하는 등 해당 프로그램을 실행하기 위한 컴퓨팅 환경을 준비해야 한다. 그러나 코랩을 이용하면 프로그램을 설치할 필요 없이 누구나

웹 브라우저를 통해 임의의 파이썬 프로그램 코드를 작성하고 실행할 수 있다. 코랩에도 몇 가지 제약사항이 존재하지만 간단하게 데이터를 조회하기에는 충분할 것이다.

먼저 기상청 기상자료에 대한 조회를 위해 pip 명령어로 pykawos를 설치한다. pykawos와 pyrdawos은 모두 ‘read_station’ 명령어로 관측지점을 쉽게 조회할 수 있다. Fig. 3은 기상청 종관기상관측지점의 history와 현재까지도 관측하고 있는 지점 리스트를 조회하는 방법을 보여준 그림이다.

기상청 종관기상관측자료와 방재기상관측자료를 이용해 최근 연평균 기온의 변동을 조회하고자 한다. Fig. 4와 같이 read_single_point 명령어를 통해 충주(127) 지점과 안성(516)지점의 2010년부터 2021년까지의 기상자료를 불러왔다. 두 지점의 11년 데이터를 불러오는 데는 약 3초의 시간이 소요되었다. 그 후 각 데이터를 groupby와 mean을 통해 연도별로 평균하였다. 그 후 matplotlib 모듈을 이용해 두 지점의 2010년부터 2021년까지의 연평균 기온 그래프를 그려주었다.

충주와 안성지역의 지난 11년간 연평균 기온의 변동은 Fig. 5와 같이 나타났다. 두 지점을 비교해보면 안성지역보다 충주지역의 연평균 기온이 더 낮은 것으로 나타났으며, 많게는 1.1°C, 적게는 약 0.2°C의 차이가 있는 것으로 보였다. 두 지역 모두 연평균 기온이 2011년부터 2016년까지 상승세를 보였으며, 이 5년간 기온이 1.2°C 이상 상승했다. 2017년과 2018년에는 기

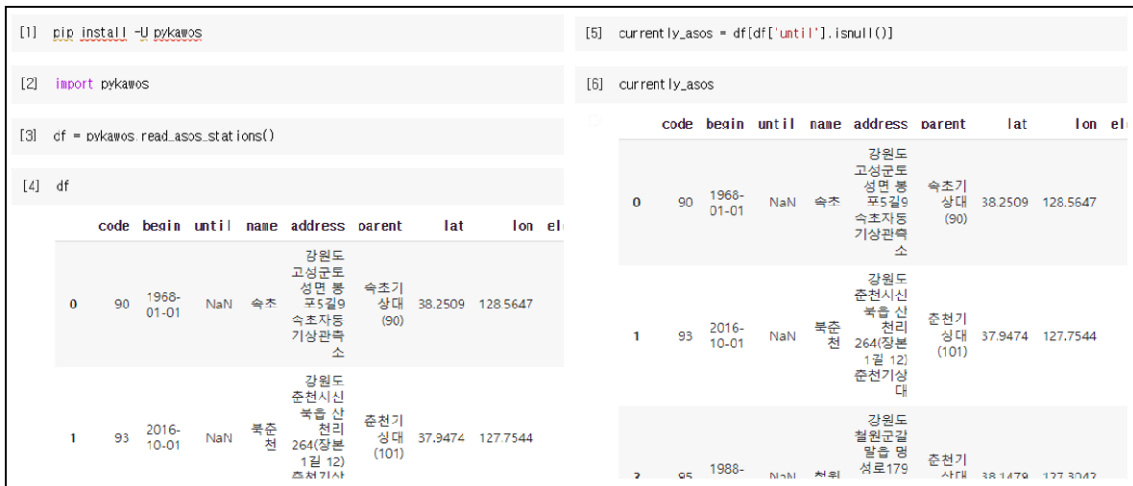


Fig. 3. A python code of ‘pykawos’ package installation and weather observation point inquiry.

```
[9] import matplotlib.pyplot as plt
[10] plt.rcParams['figure.figsize'] = [20,4]
[46] df127 = pykawos.read_asos_single_point(127, '20100101', '20211231')
[54] df127['year'] = df127['dt'].str[:4]
[55] avg_df127 = df127.groupby(['year'], as_index = False).mean()
[56] avg_df127
```

	year	index	avg_ta	min_ta	max_ta	max_ta_hrat	mi
0	2010	182.0	11.588493	6.525753	17.280274	1401.306849	
1	2011	182.0	11.438904	6.208767	17.294521	1402.369863	
2	2012	182.5	11.528415	6.410929	17.319945	1413.237705	
3	2013	182.0	11.836986	6.647123	17.670959	1390.715068	
4	2014	182.0	12.251507	6.903288	18.420822	1434.304110	
5	2015	182.0	12.726849	7.324384	18.783836	1393.301370	
6	2016	182.5	12.917486	7.507377	19.052459	1406.497268	
7	2017	182.0	12.006712	6.433154	18.350411	1414.056164	

```
[60] df516 = pykawos.read_aws_single_point(516, '20100101', '20211231')
df516['year'] = df516['dt'].str[:4]
avg_df516 = df516.groupby(['year'], as_index = False).mean()
avg_df516
```

	year	index	ta	tn	tx	rn	rx_1f
0	2010	182.0	12.715110	7.989863	17.856712	4.072603	1.64931
1	2011	182.0	11.968493	7.310137	16.964384	5.034247	1.70914
2	2012	182.5	12.052186	7.426503	17.042623	3.816940	1.43442
3	2013	182.0	12.495278	7.714247	17.626027	3.343407	1.37912
4	2014	182.0	12.475068	7.193425	18.579178	2.927397	0.91232
5	2015	182.0	12.879396	7.669863	18.919178	2.404110	0.96301
6	2016	182.5	13.170492	8.031694	19.192623	1.923497	Na

```
[63] line1 = plt.plot(avg_df516['year'], avg_df516['ta'], 'bo--')
line2 = plt.plot(avg_df127['year'], avg_df127['avg_ta'], 'rd--')
plt.xticks(np.arange(11, 14, 0.5))
plt.tick_params(axis = 'both', labelsize = 14)
plt.xlabel('Year', labelpad = 15, fontsize = 18)
plt.ylabel('Mean temperature(°C)', labelpad = 15, fontsize = 18)
plt.legend(handles = (line2, line1), labels = ('Chungju(127)', 'Anseong(516)'),
loc = 'upper left', ncol = 2, fontsize = 15)
plt.savefig('fig1.png', dpi=300)
```

Fig. 4. A python code of weather data inquiry and code for graph display using 'pykawos'.

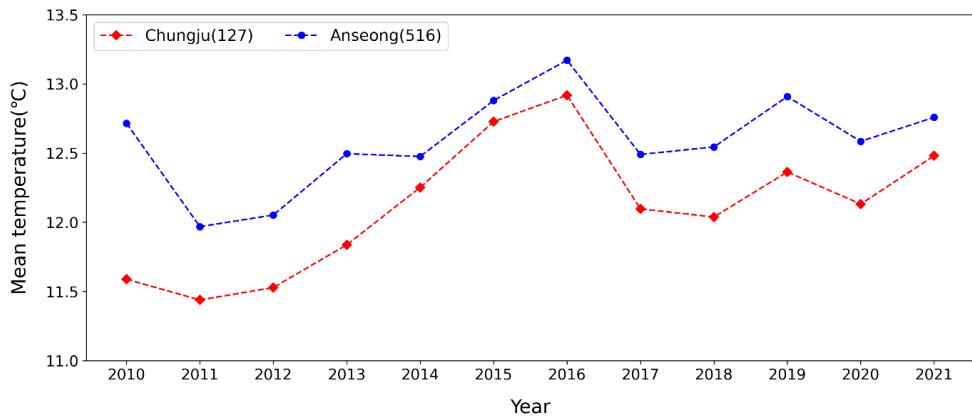


Fig. 5. Trends of average temperature in Chungju and Anseong for 11 years.

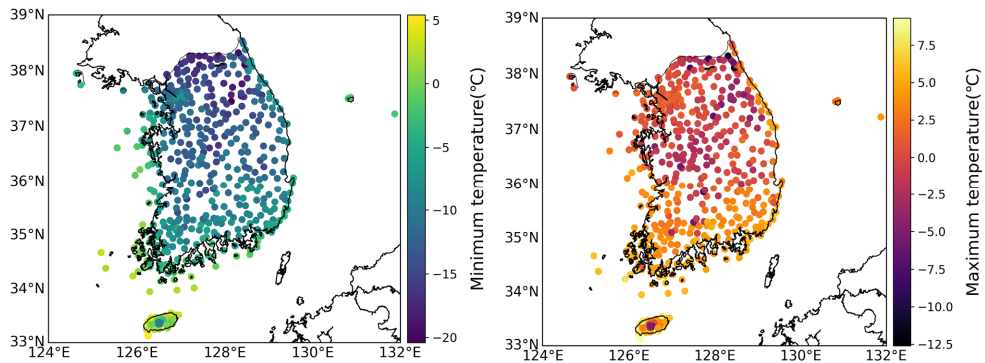


Fig. 6. Scatter plot of daily minimum and maximum temperature over the Korean peninsula on January 1, 2021.

온이 약간 떨어졌다가 그 후 2021년까지 기온이 다시 상승하고 있는 것을 확인할 수 있었다. 두 지점 모두 기온이 가장 높았던 것으로 나타난 2016년은 전 지구의 평균기온이 역대 최고로 높았던 해로, 우리나라 평균기온 역시 평년(12.5°C)보다 1.1°C 높아 1973년 이래 최고 1위를 기록한 해였다. 이때 폭염 및 가뭄으로 인하여 16,599.7ha의 농작물 피해가 발생하기도 했다 (KMA, 2016).

이번엔 특정 날짜에 대한 모든 관측지점의 값을 한번에 반환하는 `read_multi_point` 명령어를 이용해 2021년 1월 1일 기상청 방재기상관측지점의 기상자료를 조회했다. 관측값을 사용자에게 반환할 때 관측지점의 위치값도 함께 반환하므로 그림과 같이 간단한 명령어를 통한 시각화가 가능하다.

IV. 고찰 및 결론

본 연구에서는 농업연구자들의 기상자료 접근성 및 활용성을 높이고자 기상자료를 모아 원격저장소 호스팅 서비스인 깃허브에 업로드하고, 이를 쉽게 가져올 수 있는 파이썬 패키지를 제작하였다. 그 후 구글 코랩을 통해 기상청 종관기상관측과 방재기상관측 두 지점의 2010년부터 2021년까지 총 11년의 자료를 조회하고 간단한 변동을 살펴보는 작업을 통해 패키지의 편의성을 평가하였다. 이 자료를 기상자료개방포털에서 조회하고자 한다면 ‘종관기상관측’, ‘방재기상관측’ 탭 각각에서 csv나 excel 파일로 자료를 다운로드 해 확인해야 하고, 일기상관측자료의 경우 한 번에 최대 10년씩만 조회할 수 있어 각각 두 번씩 총 네 번의 조회작업을 거쳐야한다. 본 연구에서 제작한 패키지를 이용하면 구글 코랩 내에서 간단한 명령어로 한 번에 자료를 조회할 수 있으며, 자료를 다운로드 하는 데 걸린 시간은 3초가 채 되지 않았다. 불러온 자료는 다시 파일을 열거나 하는 작업 없이 곧바로 이용할 수 있다. 만일 특정 일자의 모든 관측지점 값을 조회하기 위해서는 늘어난 지점 수만큼 자료 다운에 더 많은 시간이 소요된다. 본 논문에서는 간단한 명령어로 짧은 시간에 전체 관측지점의 자료를 조회한 후 2021년 1월 1일의 전국 최저기온 및 최고기온을 시각화했다.

기상자료와 같은 공공데이터의 경우 공공기관의 사이트나 API로 제공하는 자료는 ‘공신력’을 가지고 있다. 본 패키지를 이용해 제공하는 자료가 이러한 정도의 자료 신뢰성, 즉 공신력에 도전하는 것은 어려울

것으로 보인다. 그러나 본 논문의 패키지는 사용자의 기상자료 이용 편의성을 최대화하기 위해 추가적인 인증 절차나 환경 구축이 따로 필요하지 않고 누구나 파이썬 패키지를 설치하는 것만으로 이용이 가능하도록 설계되었다. 따라서 대량의 기상자료가 필요한 작업이나 기상자료를 저장한 기기가 없는 곳에서도 소프트웨어 도구를 이용 가능한 기기만 있다면 언제 어디서든 연구 및 분석이 가능하고, 기상관측지점 조회나 기상자료를 이용한 간단한 모델 구동 등이 더욱 간편해질 것이다. 또한 굳이 사용자가 기상자료를 저장해 두기 위한 공간을 따로 할당하지 않아도 된다. 웹에 업로드된 자료를 언제든지 불러와 사용할 수 있기 때문이다. 농촌진흥청의 경우 현재에는 운영하지 않는 관측지점의 기상자료는 더 이상 농업날씨365를 통해 조회할 수 없게 된다. 이 경우 역시 깃허브에 자료가 저장되어 있으므로, 언제든지 과거 관측지점의 자료를 조회할 수 있을 것이다. 기상자료는 모두 기상청과 국립농업과학원에서 공공누리의 제1유형으로 공개한 자료임을 확인했다. 공공누리란 공공저작물 자유이용 허락 표시제도(Korea Open Government License)로 공공기관이 4가지 유형 마크를 통해 개방한 공공저작물 정보를 통합 제공하는 서비스이다. 이 중 제1유형은 출처를 표시할 경우 상업적, 비상업적 이용이 가능하고 변형 등 2차적 저작물 작성 또한 가능하다.

수집한 기상자료와 제작한 패키지는 모두 깃허브의 공개 저장소에 출처 표시와 함께 업로드했다. 깃허브는 분산 버전 관리 시스템인 깃을 기반으로 한다. 깃의 분산 버전 관리 기능은 개인이 저장소 전체를 복사해 개별 작업이 가능하고 원격 저장소를 통해 여러 참여자의 자료 수정 및 개발이 가능하다. 또한 버전 이력을 관리할 수 있어, 소스 코드의 변경 사항을 확인하거나 이전 상태로 되돌릴 수 있다(version control, 2021). 깃허브에서 제공하는 포크(fork) 기능은 공식적으로 프로젝트의 재사용이 허가된 오픈 프로젝트를 자신의 계정의 원격 저장소에 그대로 복사할 수 있는 기능이다(Kang et al., 2017). 이 기능을 이용해 타인의 프로젝트를 자신의 저장소에서 수정할 수 있으며, 원래의 프로젝트 주인에게 수정사항 반영을 요청할 수도 있다. 본 연구에서는 기상자료 재배포 패키지에 대한 가능성을 제시하는 것으로 아직 개선해야 할 점이 존재한다. 현재 깃허브에 저장한 기상청 기상자료는 2021년까지의 일 기상관측 자료이고, 농촌진흥청 기상자료는 2022년 8월까지의 일 기상관측자료이다. 농업 연구

에 다양하게 쓰이기 위해서는 시간별 기상관측자료가 추가되어야 할 것이며, 현재 시점까지의 기상자료를 꾸준히 업데이트할 필요가 있다. 깃허브의 기능을 이용하면 여러 사용자를 통해 이러한 자료 자체에 대한 개선뿐 아니라 기능에 대한 개선까지 자유롭게 이뤄질 것으로 기대된다.

기상청 및 농촌진흥청에서 제공되는 자료 품질 자체에 대한 아쉬움 역시 존재한다. API를 통해 수집한 농촌진흥청 기상자료의 경우 약 37개 지점의 자료를 확보할 수 없었다. 또, 농촌진흥청에서 제공하는 강우량 자료의 품질이 의심되는 경우가 있었다. 예를 들어 농업날씨365에서 제공하는 시간별 자료에는 2020년 8월 10일, 안성시 보개면에 17시부터 22시까지 시간마다 16.5mm 이상의 비가 내렸다고 기록되었으며, 23시에는 31.5mm의 강우가 기록되었다. 같은 날, 일 강우량 기록에 따르면 안성시 보개면에 내린 총 강우량은 31.5mm이다. 같은 지점의 2020년 8월 3일 기록에는 13시부터 23시까지 10시간 동안 시간별 강우량이 100mm가 넘는다. 따라서 추후 시간별 자료를 추가할 경우 자료 자체의 품질에 대한 관리 역시 고민할 필요가 있을 것이다.

적 요

농업은 기상에 매우 민감한 산업으로, 따라서 농업 분야의 기상을 이용한 연구는 더욱 중요해지고 있다. 연구자들은 기상청과 농촌진흥청에서 제공하는 기상 정보서비스 웹사이트에 접속해 기상관측자료를 다운로드할 수 있다. 그러나 대량의 기상자료를 받아야 할 때는 여러 번의 조회작업이 필요한 단점이 있다. 본 데이터 논문은 기상청과 농촌진흥청에서 수집한 자료를 원격 저장소 서비스인 깃허브에 업로드하고 소프트웨어 프로그램인 파이썬을 이용해 기상자료에 쉽게 접근할 수 있는 패키지를 제작했다. 이를 통해 추가적인 인증 절차 없이 누구나 자료를 가져갈 수 있는 방식을 채택하여 농업 관계자들의 기상자료에 대한 접근성 및 활용성을 높이는 방법을 제안한다. 자료와 패키지는 분산 버전 관리 시스템인 깃에 업로드하여 수정 및 관리가 용이하게 하였다.

감사의 글

본 연구는 농촌진흥청 국립농업과학원 농업과학기

술 연구개발사업 (과제번호: PJ0148942022)의 지원에 의해 수행되었음.

REFERENCES

- Cho, Y. N., C. Y. Yoon, H. K. Kim, H. D. Moon, K. N. An, and J. I. Cho, 2020: Meteorological data measured under Agrivoltaic Systems in Boseong-gun during winter barley season. *Korean Journal of Agricultural and Forest Meteorology* **22**(3), 144-151. (in Korean with English abstract)
- Hur, J. N., J. H. Park, K. M. Shim, Y. S. Kim, and S. R. Jo, 2020: A high-resolution agro-climatic dataset for assessment of climate change over South Korea. *Korean Journal of Agricultural and Forest Meteorology* **22**(3), 128-134. (in Korean with English abstract)
- Jang, Y. B., I. H. Jang, and Y. C. Choe, 2020: Prediction of soil moisture with open source weather data and machine learning algorithms. *Korean Journal of Agricultural and Forest Meteorology* **22**(1), 1-12. (in Korean with English abstract)
- Jeong, H. K., J. H. Sung, and H. J. Lee, 2020: Analysis of Social Demand for Countermeasures in Response to Extreme Weather Events in Korean Agricultural Sector. *Journal of Climate Change Research* **11**(4), 235-246. (in Korean with English abstract)
- Kang, H. S., J. H. Cho, and H. C. Kim, 2017: *Case Study on Software Education using Social Coding Sites*. *Journal of Digital Convergence* **15**(5), 37-48. (in Korean with English abstract)
- Kim, H. K. H. D. Moon, Y. N. Cho, S. H. Sin, J. H. Kim, Y. W. Lee, and J. I. Cho, 2021: Direct and Diffuse Radiation Data in Naju During May 2019 to November 2020. *Korean Journal of Agricultural and Forest Meteorology* **23**(2), 134-140. (in Korean with English abstract)
- Korea Meteorological Administration (KMA), 2017: Abnormal Climate Report 2016. KMA, 58pp. (in Korean)
- Korea Meteorological Administration (KMA), 2021: Abnormal Climate Report 2020. KMA, 78-81. (in Korean)
- Korea Meteorological Administration (KMA), 2022: Abnormal Climate Report 2021. KMA, 78-80. (in Korean)
- Lee, J. H., 2016: 공학적 데이터 처리를 위한 파이썬 (Python) 언어의 활용: Spyder (Scientific PYthon

- Development EnviRonment). *The Korean Institute of Electrical Engineers* **65**(5), 41-48.
- Lee, S. H., and I. Y. Lee, 2011: Effective Searchable Symmetric Encryption System using Conjunctive Keyword on Remote Storage Environment. *The KIPS Transactions:PartC* **18C**(4), 199-206.
- Noh, H. J., and J. H. Son, 2021: 오픈 API 기반의 금융생태계 변화와 시사점. KIRI report(포커스). **521**, 8-13.
- Park, J. H., Y. S. Shin, and K. M. Shim, 2021: Improvements of Unit System for nationwide expansion of Early Warning Service for Agrometeorological Disaster. *Korean Journal of Agricultural and Forest Meteorology* **23**(4), 356-365. (in Korean with English abstract)
- Shim, K. M., Y. S. Kim, M. P. Jung, J. W. Kim, M. S. Park, S. H. Hong, and K. K. Kang, 2018: Recent Changes in the Frequency of Occurrence of Extreme Weather Events in South Korea. *Journal of Climate Change Research* **9**(4), 461-470. (in Korean with English abstract)
- Version control, 2021: https://en.wikipedia.org/wiki/Version_control (2021. 4. 15. Accessed)