

Matching spatial temporal data of forest fire with meteorological information

Date: 24, July 2012

Presenter: Sol-E Choi

Contents



Introduction



Advantage of Python



Method



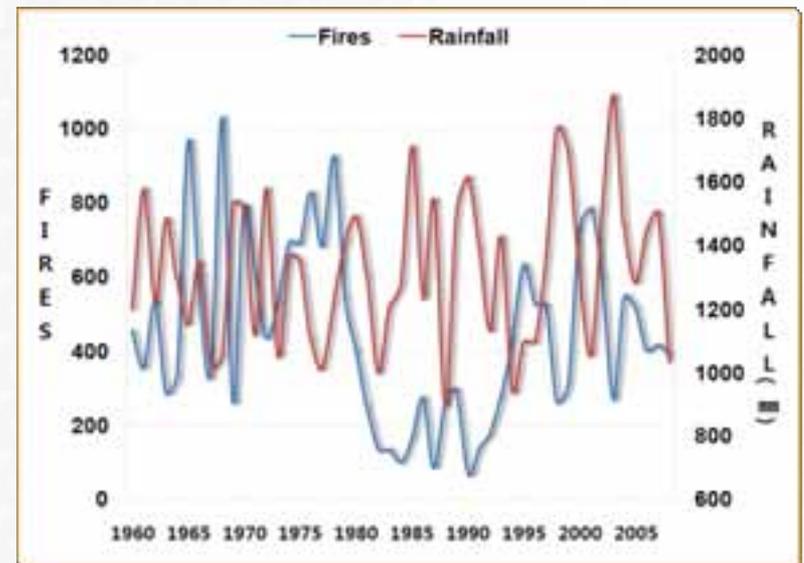
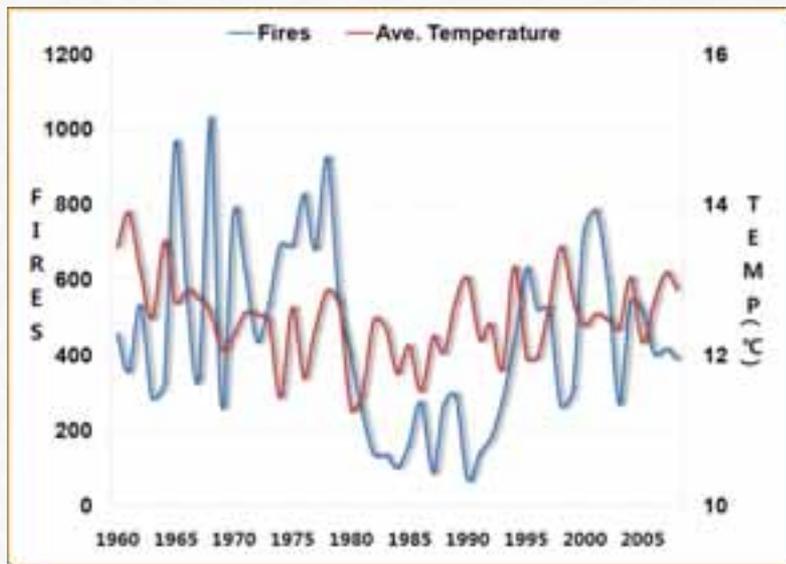
Result



Conclusion

1. Introduction

- Climate and weather are of extreme importance for the ignition and propagation of forest fires.
- Fire characteristics such as temperature, intensity and duration, as well as season and frequency of fire, have important effects on vegetation response to fire.



1. Introduction

- The Intergovernmental Panel on Climate Change report (IPCC, 2007) suggests that, with global warming, forest fires frequency will increase all over the world.
- Several studies point that global warming is likely to increase fire frequency and severity
- Warmer and drier conditions will increase the forest fire risk in the region.
- In Korea, where weather has been found to be the most important factor influencing forest fires
- This work is focused in matching forest fire data to meteorological factors which obtained by weather stations using Python script and IDW interpolation.
- The forest fire history which consists of specific point of fire location, occurred date, occurred time and reason was collected by Korea Forest Research Institute.

1. Introduction

● High Temp., Dryness on Climate impact recently

- High Temperature and continuous drought phenomenon in winter
 - Increase forest fire in Fall/Winter
- Increase forest fire occurrence potential by early agricultural activities

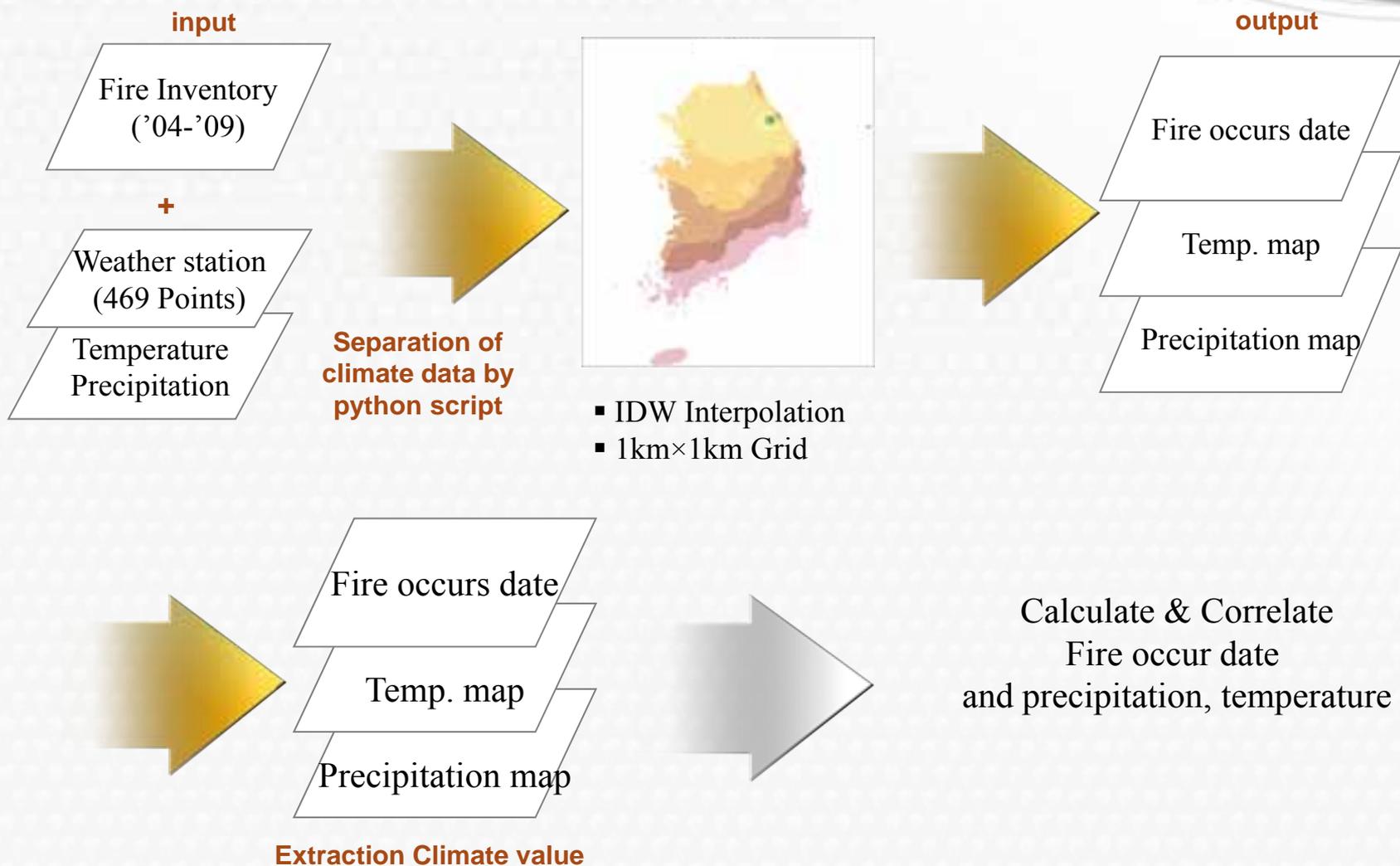
● Climate Change on Global warming

- Increase possibilities of large-sized forest disaster such as forest fire, landslide and forest disease/insect. etc.
- Needs to develop social disaster prediction techniques

2. Advantage of Python

- With the addition of the ArcPy module in ArcGIS 10 there is so much more that can be done with Python.
- If you use python, saves a lot of time is to use ModelBuilder and create a model for the desired process.
- Therefore python useful of mass data processing

3. Method



Separation of the data using by python



Separation of the data using by python

Aws point site matching Coordinate X,Y



Fig. location of aws site

Between 2005-2009 years separate daily data from aws climate data by using python script

	A	B	C	D	E	F	G	H
1	STN_ID	TM_YEAR	TM_MON	TM_DAY	RP_DAY	TA_DAY	X_COORD	Y_COORD
2	116	2005	1	1	0	-0.9	126.967	17.4433
3	144	2005	1	1	0	-2.9	126.788	16.9097
4	160	2005	1	1	0	-6.2	129.0021	15.1157
5	300	2005	1	1	0	0.6	126.3167	15.85
6	301	2005	1	1	0.5	2.7	126.1111	15.08389
7	302	2005	1	1	2	1	126.1333	14.6
8	303	2005	1	1	0	2.4	125.1167	14.0667
9	304	2005	1	1	0	1.5	128.8272	14.1189
10	305	2005	1	1	0	1.5	128.9167	13.9
11	306	2005	1	1	0	-0.9	127.7833	14.4333
12	307	2005	1	1	0	1.1	127.35	14.2833
13	311	2005	1	1	0	-7.1	128.0083	15.7667
14	312	2005	1	1	0	-7	129.1167	16.45
15	313	2005	1	1	0	0.8	126.75	14.9
16	314	2005	1	1	0	-13.9	127.75	15.8667
17	315	2005	1	1	0	-10.2	127.4919	15.79944
18	316	2005	1	1	0	-9.3	127.0167	15.1167
19	317	2005	1	1	0	-4.2	127.05	15.7167
20	318	2005	1	1	0	-12.2	128.6797	17.64
21	319	2005	1	1	0	0.2	130.8706	17.5075
22	320	2005	1	1	0		128.5208	18.325
23	321	2005	1	1	0	-9.3	128.2153	18.115
24	322	2005	1	1	0	-11.2	127.6736	18.2778
25	323	2005	1	1	0	-12.4	127.5540	18.2728
26	324	2005	1	1	0	-8.9	128.0667	16.8667

Fig. Logic of python script

Using python read daily data line format. To Compare two lines date, and if the different date create new file that is contained before line daily data

A screenshot of a text file containing daily data for AWS sites. The data is organized into columns for STN_ID, TM_YEAR, TM_MON, TM_DAY, RP_DAY, TA_DAY, X_COORD, and Y_COORD. A blue arrow points to a specific line, indicating the separation of data.

Fig. separate daily temperature and precipitation data

Separation of the data; python script

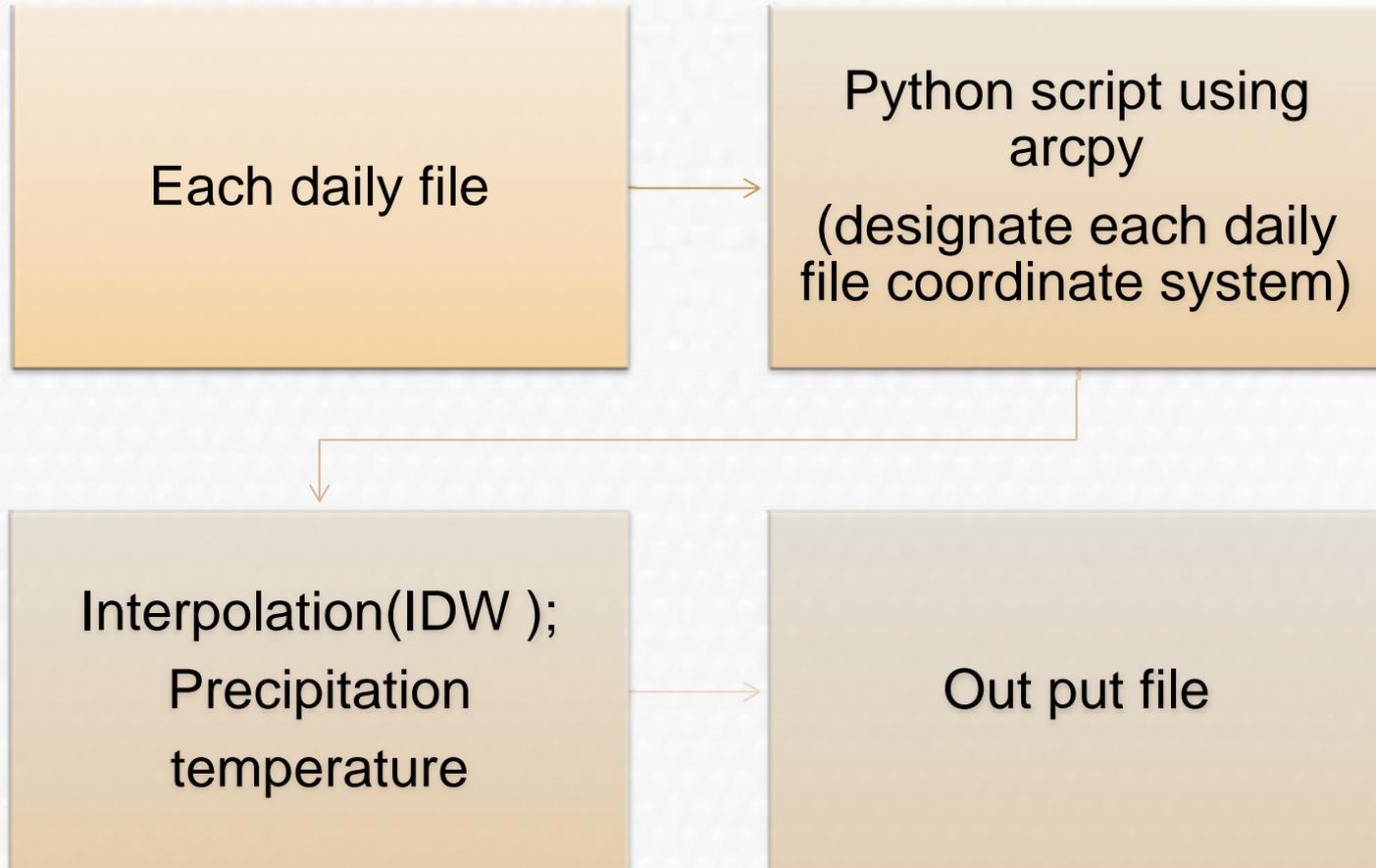
```
# -*- coding: cp949 -*-
#csv -> seperate by day.

# load csv file.
inputcsv=r"D:\00 논문작업\19_ESRI UC 2012\awswhole_2.csv"
f=file(inputcsv, 'r')

fldName=f.readline()

line=f.readline()
sepline=line.split(',')
ymd_before=sepline[1]+sepline[2].zfill(2)+sepline[3].zfill(2)
while line:
    if 'ymd_after' not in globals().keys():
        ymd_after="20050101"
        fname=ymd_after+".csv"
        outfile=file(fname, 'w')
        outfile.writelines(fldName)
        while line:
            outfile.writelines(line)
            line=f.readline()
            sepline=line.split(',')
            ymd_after=sepline[1]+sepline[2].zfill(2)+sepline[3].zfill(2)
            if(ymd_before != ymd_after):
                ymd_before=ymd_after
                break
        outfile.close()
```

Interpolation(IDW)



Interpolation(IDW)

Before Interpolation, Using arcpy daily data adjust projection Tokyo To WGS1984,
Daily precipitation and temperature data interpolation spatial analyst IDW 4years each day
by python script; Cell size 1000, power 2

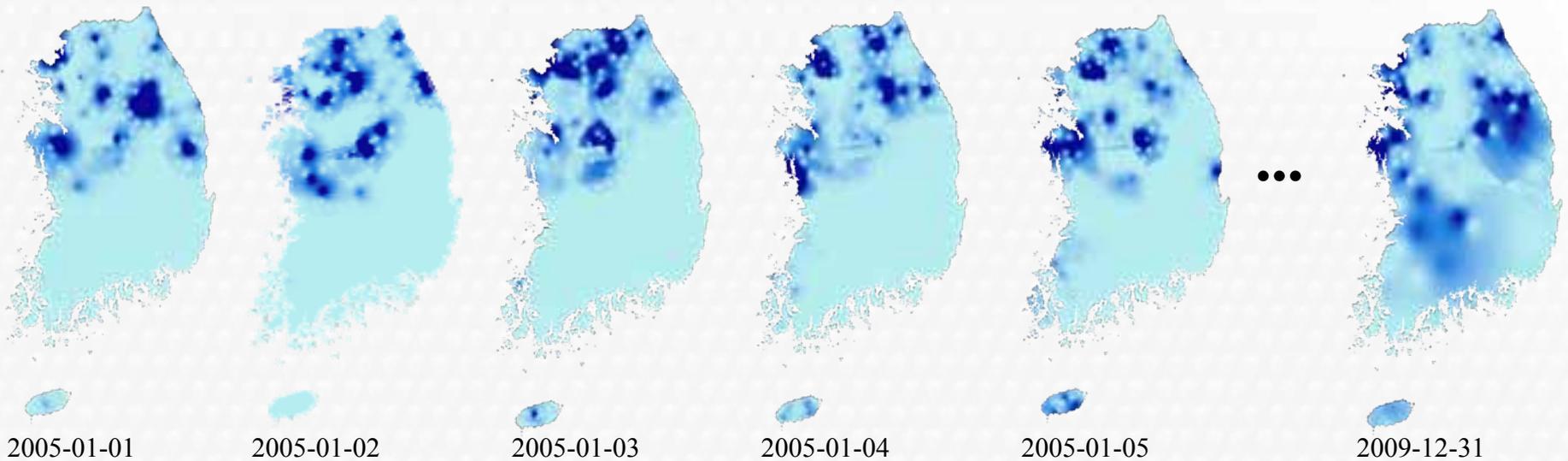


Fig. interpolation of precipitation using IDW

Interpolation(IDW); python script

```
- coding: cp949 -*-
v to XY display
p -> IDW -> Raster.

it arcpy
it os, glob

folder=r"D:\00 논문작업\19_ESRI UC 2012\aws_csv\*.csv"
older=r"D:\00 논문작업\19_ESRI UC 2012\*.csv"

t=glob.glob(in_folder)

=====
f="Coordinate Systems\Geographic Coordinate Systems\World\WGS 1984.prj"
y.OverwriteOutput = 1
=====

fname in fList:
arcpy.env.extent = "124.718724 33.117100 130.870600 38.5"
arcpy.env.workspace="D:/00 논문작업/19_ESRI UC 2012/climate"
path, name = os.path.split(fname)
basename=os.path.splitext(name)
try:
    # Set the local variables
    in_Table = fname
    x_coords = "X_COOR"
    y_coords = "Y_COOR"
    #z_coords = "POINT_Z"
    out_Layer = "d"+fname
    #saved_Layer = r"cr:\output\firestations.lyr"

    # Set the spatial reference
    spRef = r"Coordinate Systems\Geographic Coordinate Systems\World\WGS 1984.prj"

    # Make the XY event layer...
    arcpy.MakeXYEventLayer_management(in_Table, x_coords, y_coords, out_Layer, spRef, '#')

    # Print the total rows
    print arcpy.GetCount_management(out_Layer)

    # Save to a layer file
    #arcpy.SaveToLayerFile_management(out_Layer, saved_Layer)
except:
    # If an error occurred print the message to the screen
    print arcpy.GetMessages()

#좌표계 변경
try:
    arcpy.env.workspace="D:/00 논문작업/19_ESRI UC 2012/"
    outCS = arcpy.SpatialReference(r'C:\Program Files (x86)\ArcGIS\Desktop10.0\Coordinate Systems\Pr
    outfc='proj_aws/' + basename[0] + '.shp'
    arcpy.Project_management(out_Layer, outfc, outCS, transform_method="Tokyo_To_WGS_1984_5")
except:
    # If an error occurred print the message to the screen
    print arcpy.GetMessages()

#IDW 실행
```

Extract meteorological factors data

Precipitation

- Extract point data fire occurs data
- Fire inventory file add 14 field
- Extract data write each field
- Counting continuous rainless day for 2weeks

Temperature

- Correlation maximum temperature, average temperature with fire occurs date

Extract point data; python script

```
# -*- coding: utf-8 -*-
# 산물자료의 F_DATA 필드를 이용하여 14일 전까지 연속 강수량 추출

import arcpy
import datetime

arcpy.OverwriteOutput = 1

#산물 파일 불러옴
arcpy.env.workspace = "D:/00 논문작업/19_ESRI UC 2012/scl2/FF_data"
arcpy.MakeFeatureLayer_management("fire_05_09.shp", "f0509")

cur=arcpy.UpdateCursor("f0509")
arcpy.env.workspace = "D:/00 논문작업/19_ESRI UC 2012/climate"

errorlog=open("../errorlog.txt",'w')
errorlog.write("ext_climate.py error code=====\n")

for row in cur:
    date=row.F_DATE
    xcoord=str(row.COORD_X)
    ycoord=str(row.COORD_Y)
    idate=int(date)
    sdate=str(date)
    day=datetime.date(int(sdate[0:4]),int(sdate[4:6]),int(sdate[6:8])) #문자열을 day 모듈 형태로 변환
    i=0
    print idate
    for D in range(0,15): # 15일 전까지 img 불러와서 산물 속성 table D00부터 D14까지 해당 좌표의 기상값 채움
        beforeday=day - datetime.timedelta(D)
        rasterName="RN"+beforeday.strftime('%Y%m%d')+".img"
        xycoord=xcoord+" "+ycoord

        try:
            result = arcpy.GetCellValue_management(rasterName, xycoord, "#")
            Dstr="D"+str(D).zfill(2)
            row.setValue(Dstr,result)
            cur.updateRow(row)
        except:
            errorlog.write('\n-----\n')
            errorlog.write(arcpy.GetMessages().encode('utf-8'))
    # print arcpy.GetMessages()
    # print idate, rasterName, Dstr, result
    i+=1
# if idate>20050117: #시험삼아 17일까지만 돌리기
#     break

del cur, row

arcpy.CopyFeatures_management("f0509", "D:/00 논문작업/19_ESRI UC 2012/scl2/f0509_RN.shp")
```

Conclusion

- Firstly, meteorological data which includes temperature, precipitation. it was interpolated using IDW interpolation. In the case of temperature and precipitation. Secondly, these data was extracted by forest fire point according to the spatial temporal information using extraction tools in ArcGIS. These processes were done by Python Script for automation.