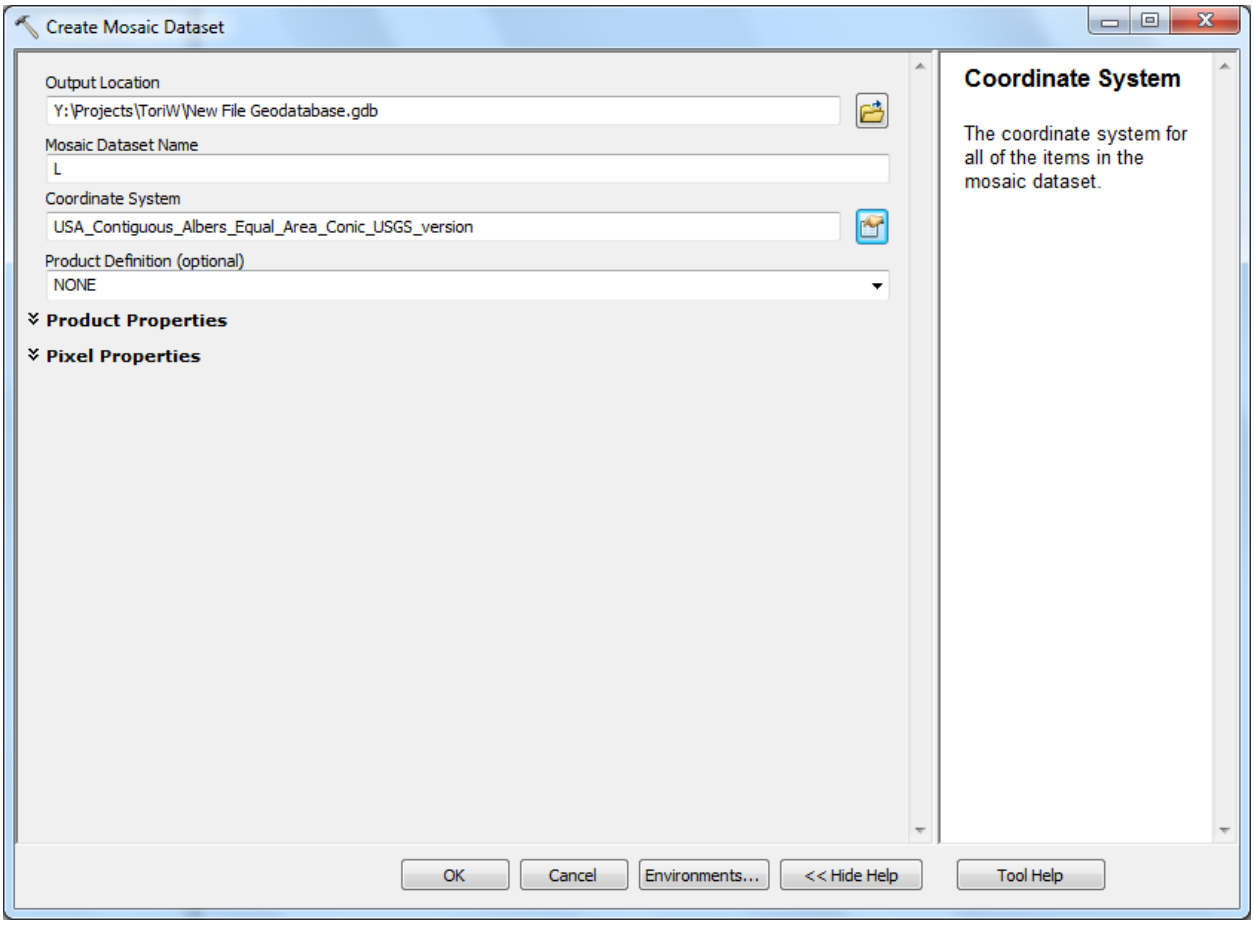
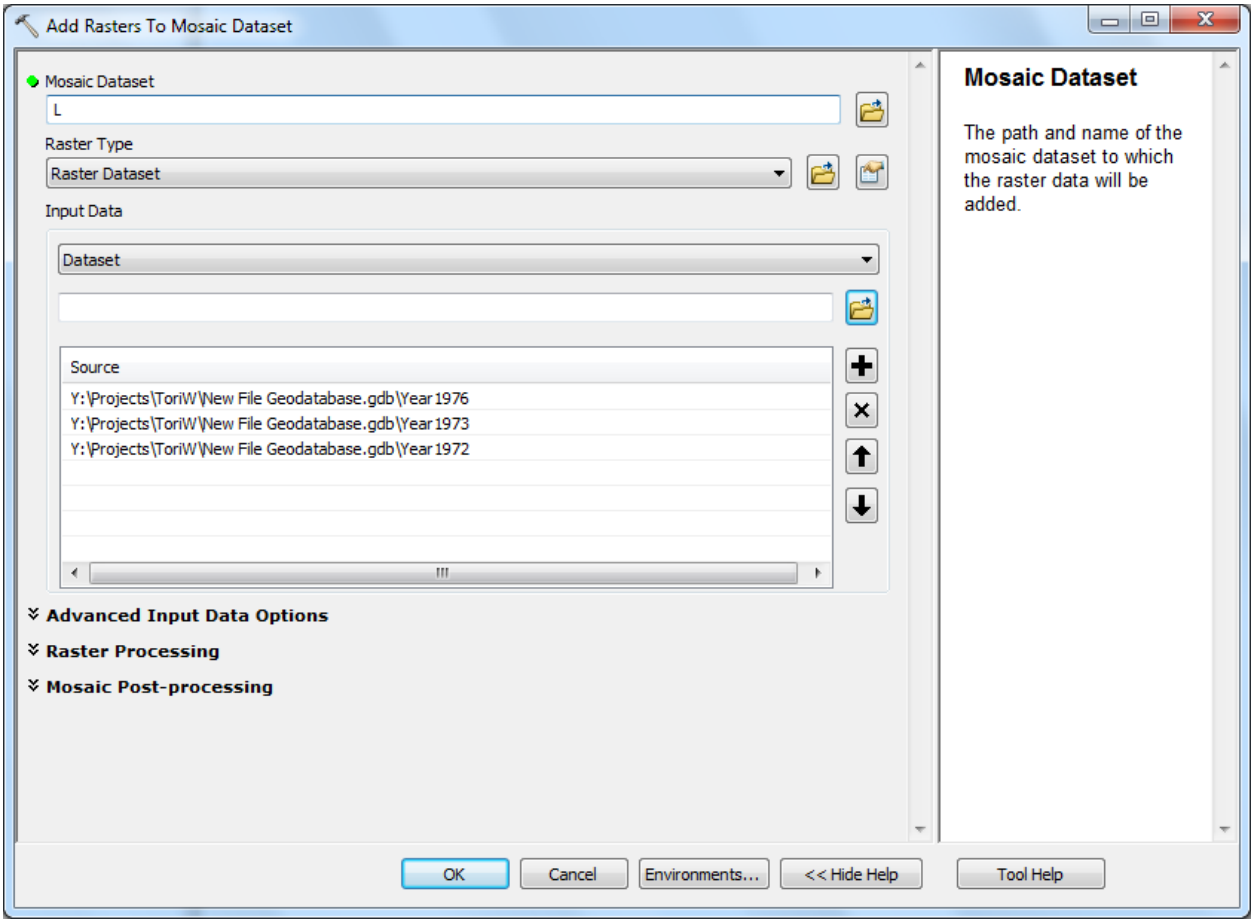
Screenshots of this Summer Work

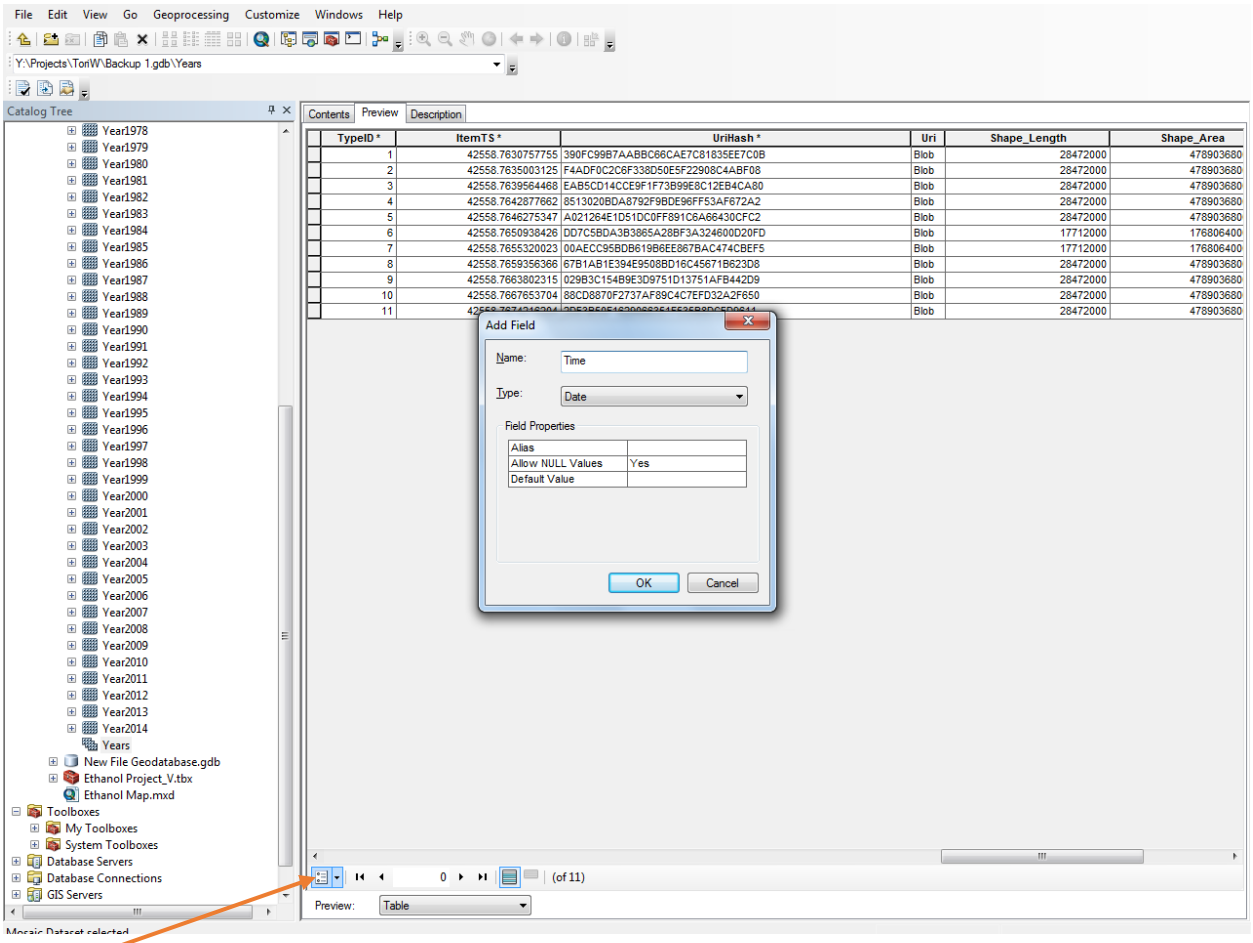
Raster Files to NetCDF File



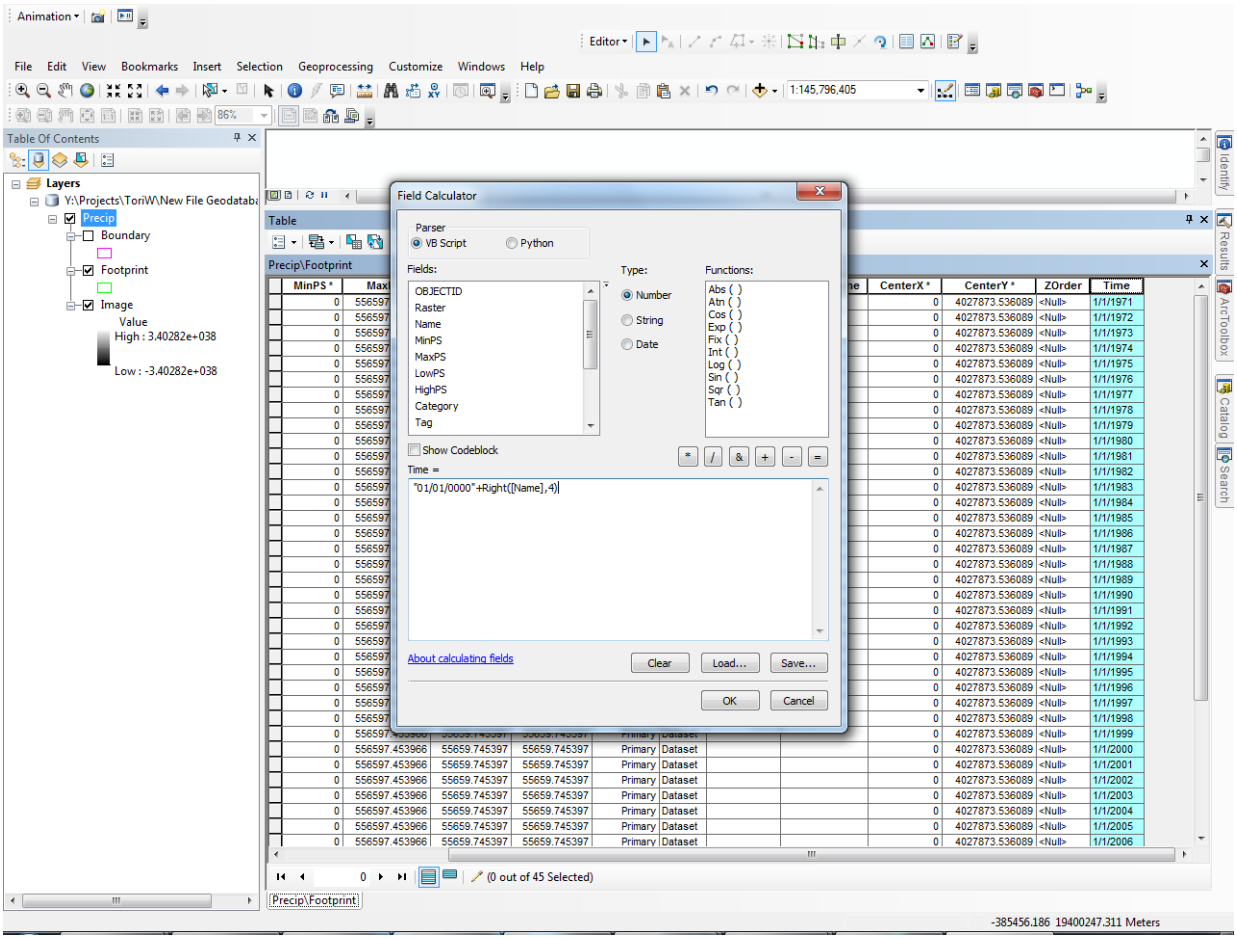
1. Create a Mosaic Dataset. Data Management Tool→ Raster→ Mosaic Dataset
    a. This allows you to insert multiple Raster files into the NetCDF, since the Raster to NetCDF only allows one file to be inserted.



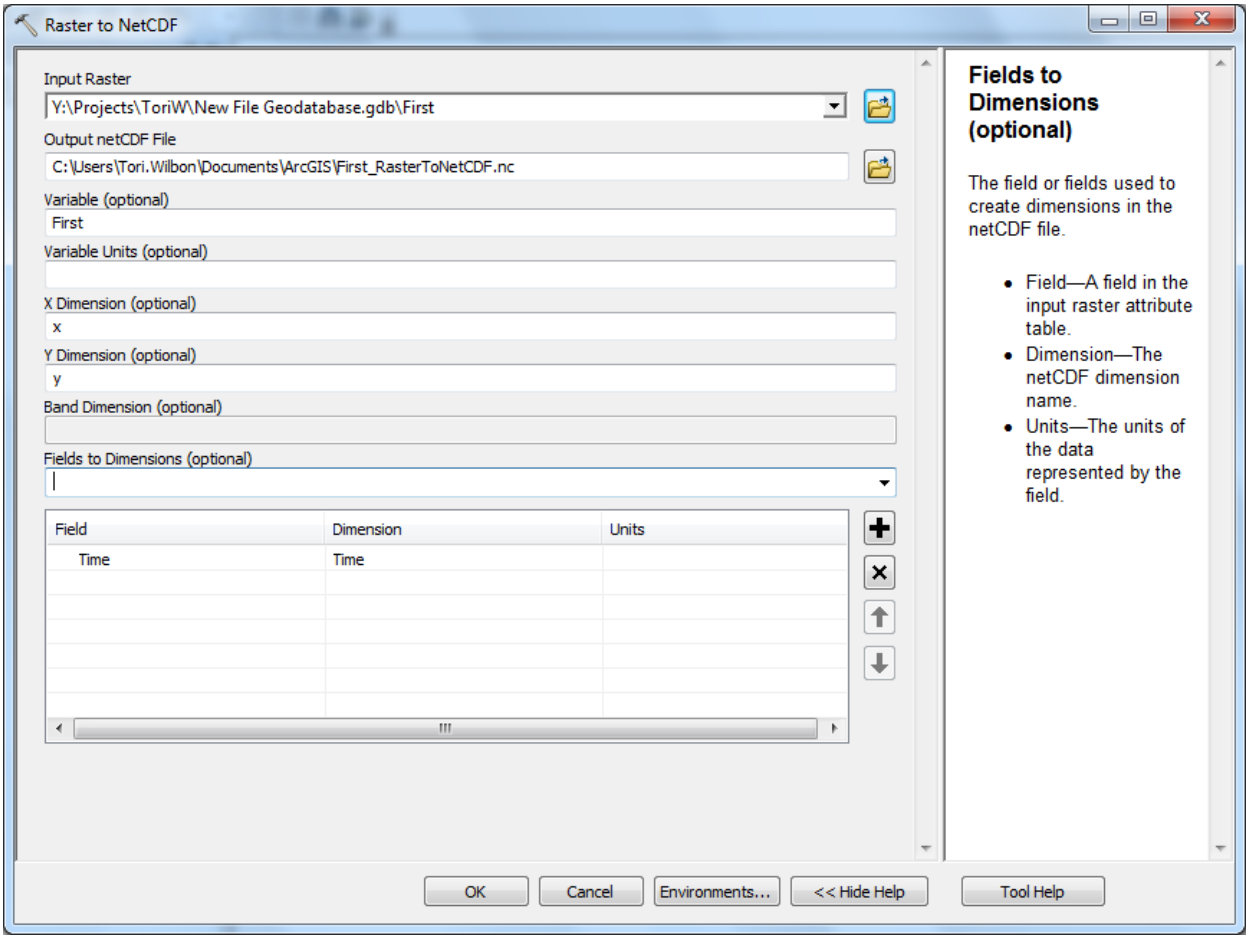
2. Add Rasters to Mosaic Dataset



3. Add Time in ArcCatalog if Time is not already there. ArcMap does not allow you to create a table to the Mosaic Dataset.
    a. Search in the Catalog Tree for the mosaic dataset, click it.
    b. Change to the preview tab.
    c. Change the preview from geography to table.
    d. Click on the icon and select add field.

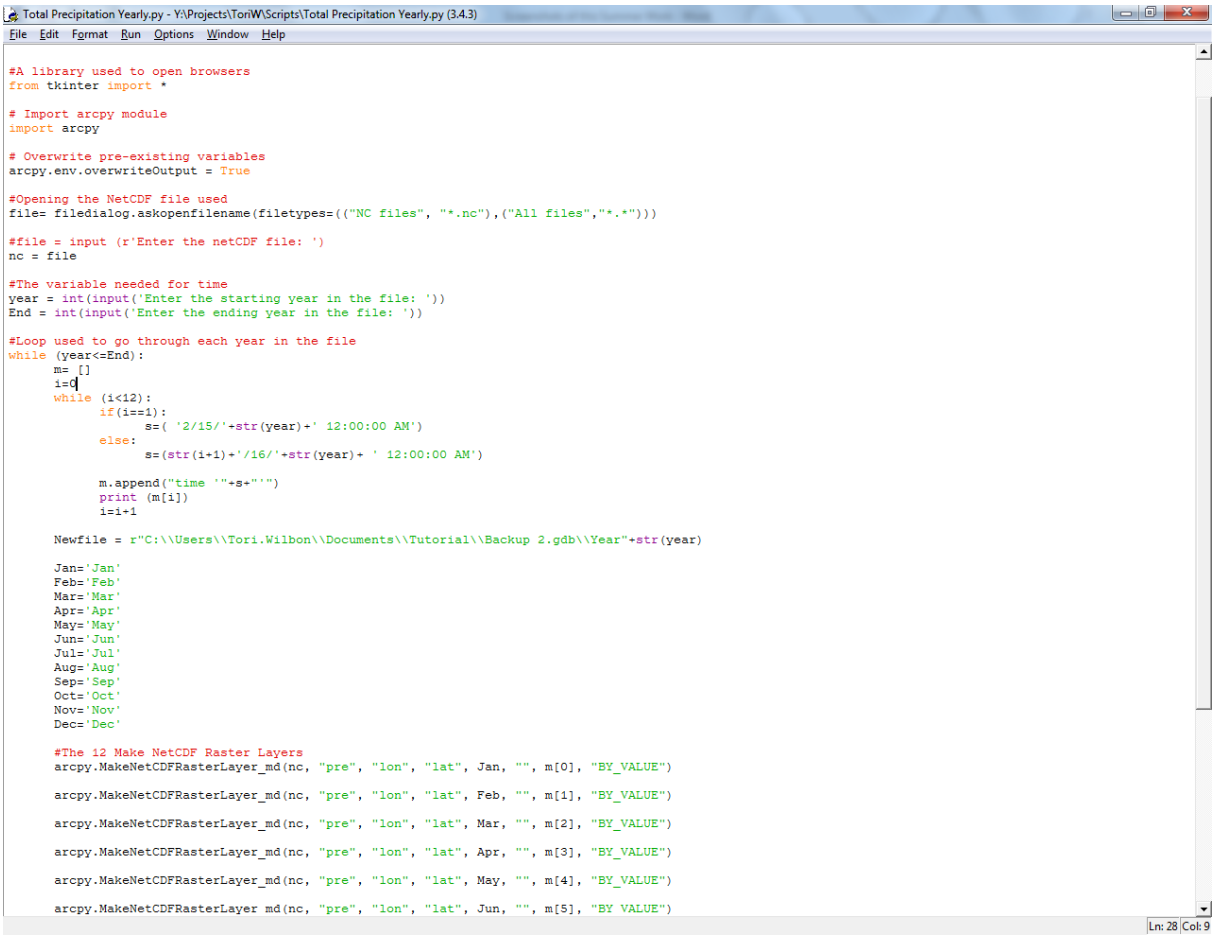


4. Populate Time Field in ArcMap using the Field Calculator formula used was “01/01/0000”+Right([Name],4) if the Name has the year inside it. The formula will change depending on what is in the Name field.



5. Create the NetCDF. Multidimension Tools→ Raster to NetCDF
    a. The units does not matter if you do not want to add them.

Taking the Total of Precipitation



6. The lines of code:
    a. Libraries that are imported are the tkinter which creates dialog boxes and arcpy
    b. The overwrite output line allows to overwrite the previous raster if something is wrong with it.
    c. This opens the NetCDF file that will be used in the code
    d. The year is the start year of the file and end is the end year of the file. This the user will need to input it.
    e. The while loop goes through each year.
    f. The m [] array will be used to hold the dates through the 12 years.
    g. Use another while loop to go through each month.
    h. Use an if else statement to see if the value is February.
    i. The Newfile is used to save the new raster which will use the year as a string to choose between them.
    j. The next Jan – Dec is needed for the Make NetCDF Raster Layer to work. If you do not have at least three characters it will not work.

k. The 12 Make Raster Layers

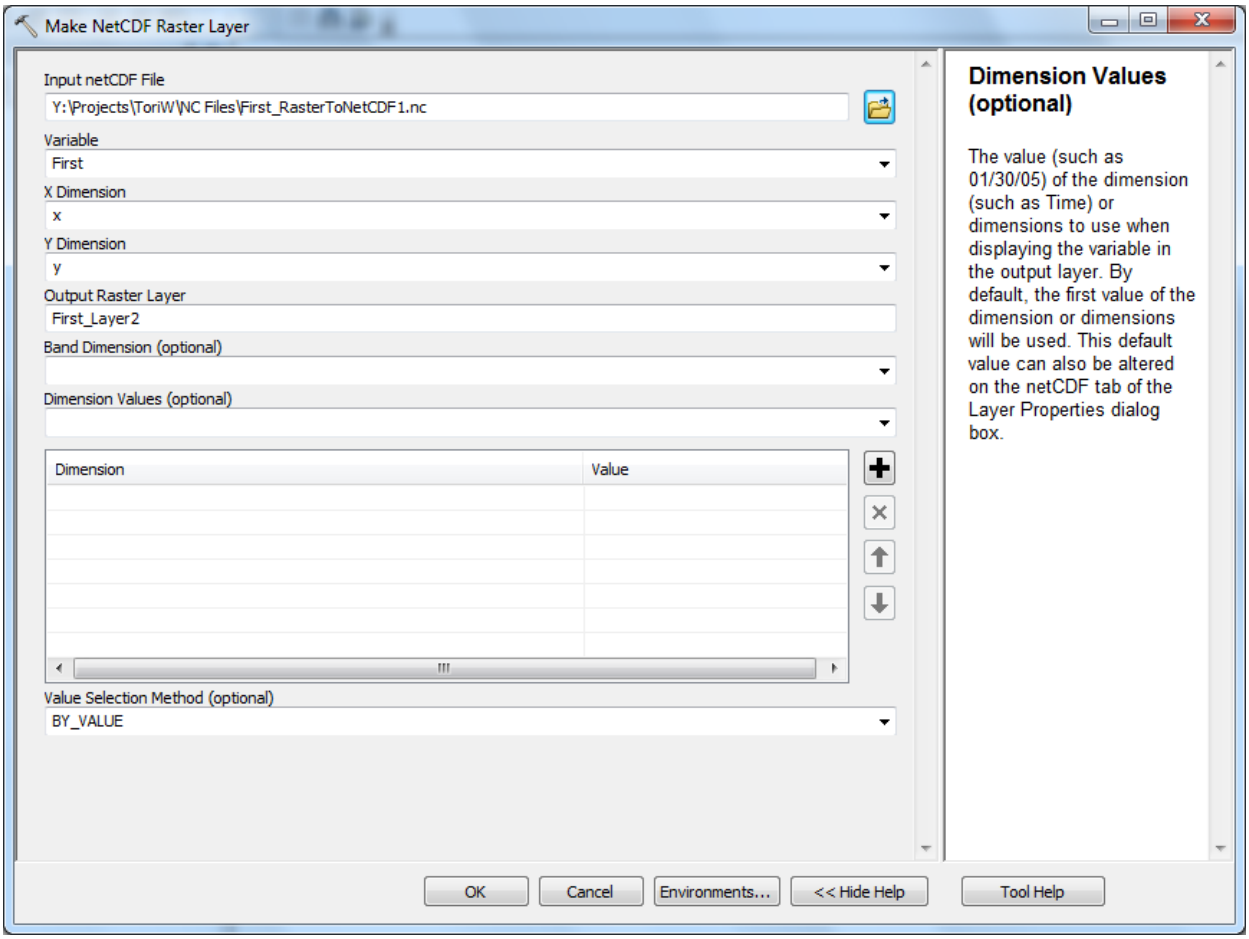
```
#The Cell Statistics which sums each year.
arcpy.gp.CellStatistics_sa("Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec", Newfile, "SUM", "DATA")

#Incrementing to the next year and Intializing"""
year=year+1
```

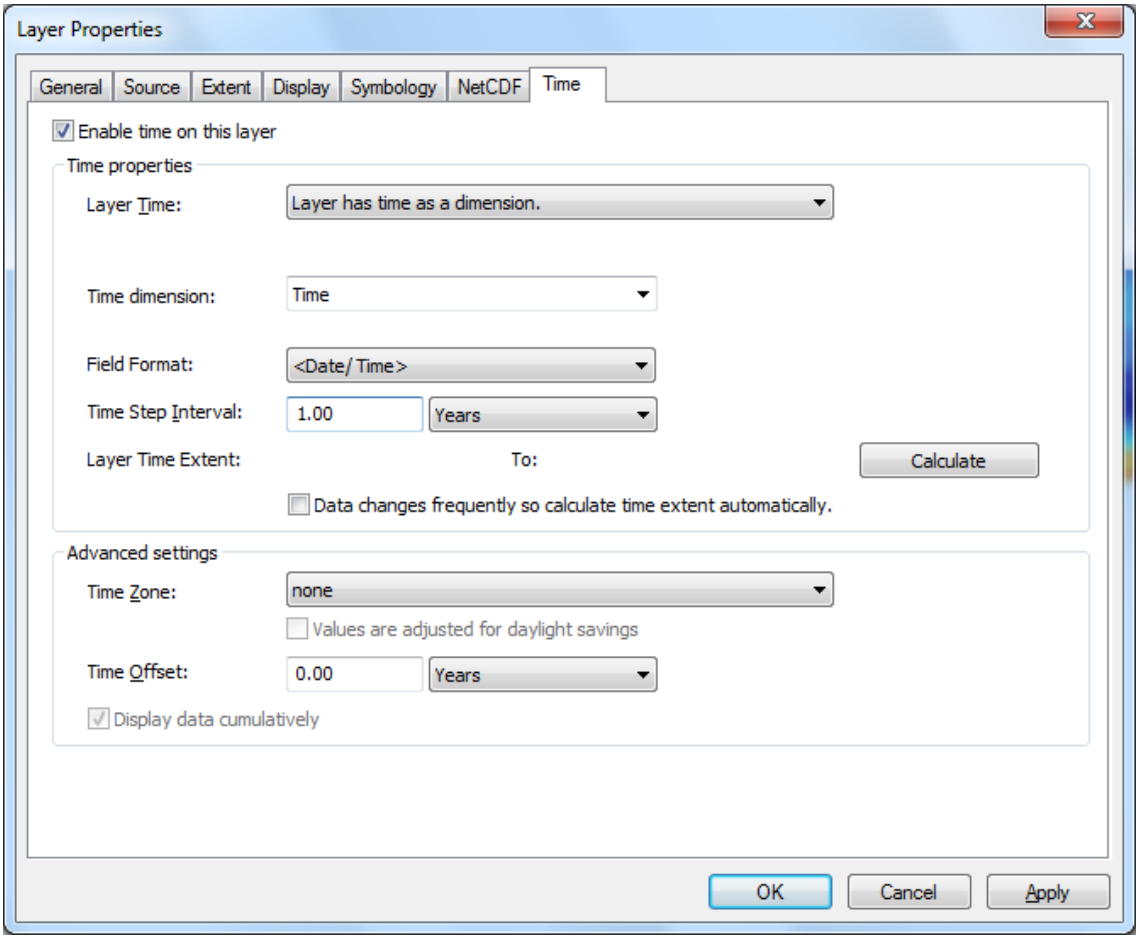
Ln: 28 Col: 9

7. This code is to find the sum of each year.
    a. Use the Cell Statistics. Cell Statistics does not like using a list/ array. This is why Jan – Dec is listed.
    b. Lastly the incrementing of the year.

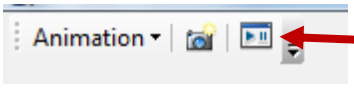
Animation



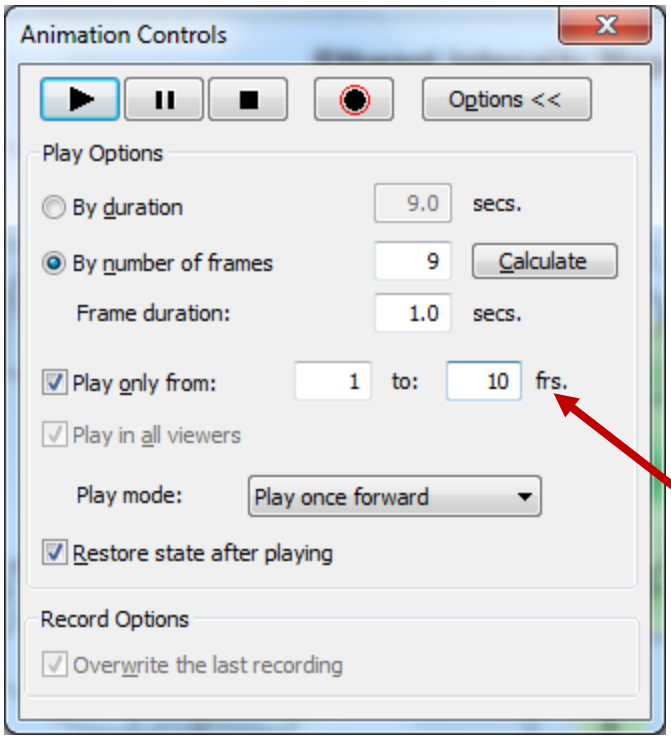
1. Change NetCDF into Raster Layer. Multidimension → Make NetCDF Raster Layer



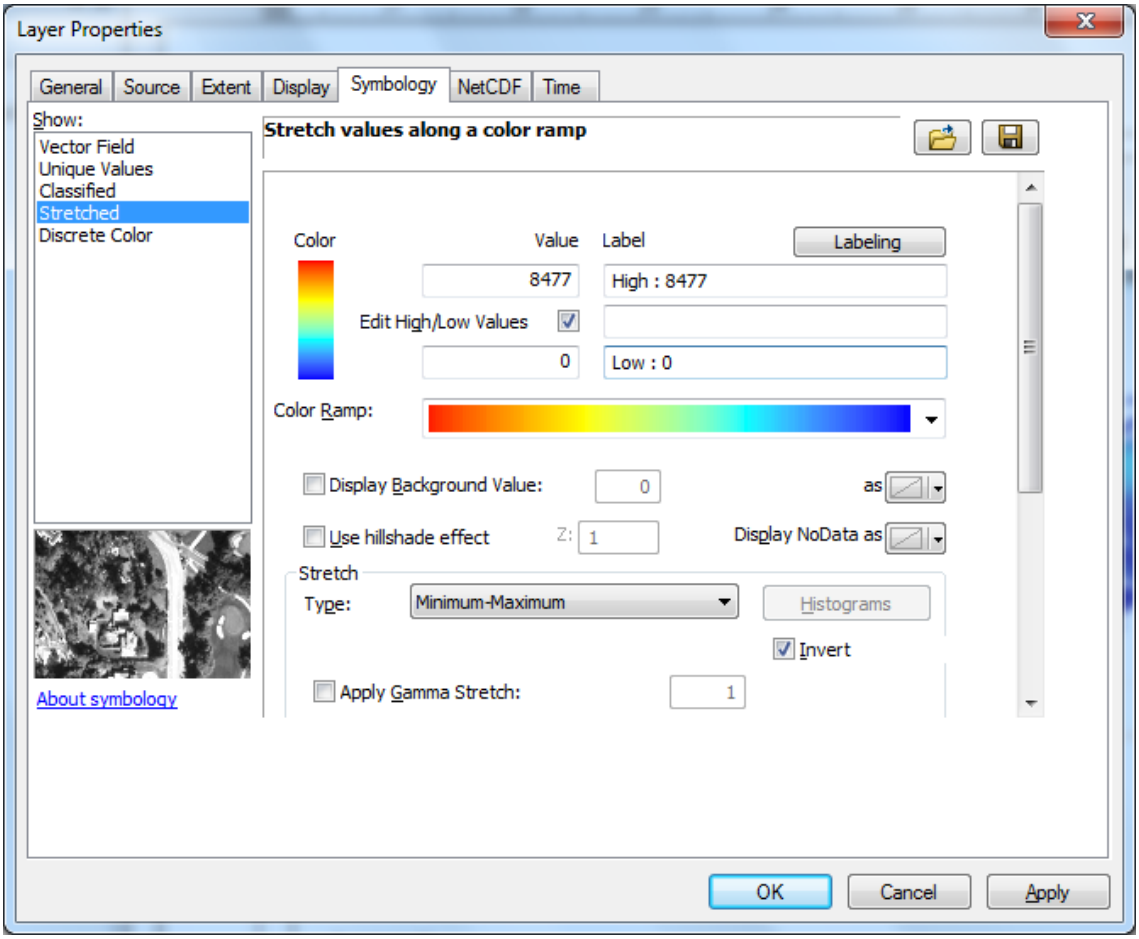
2. In properties click the Time tab and check the enable time on this layer. Choose the Time in the Time dimension and set Time Step Interval to year, month, day, or etc.



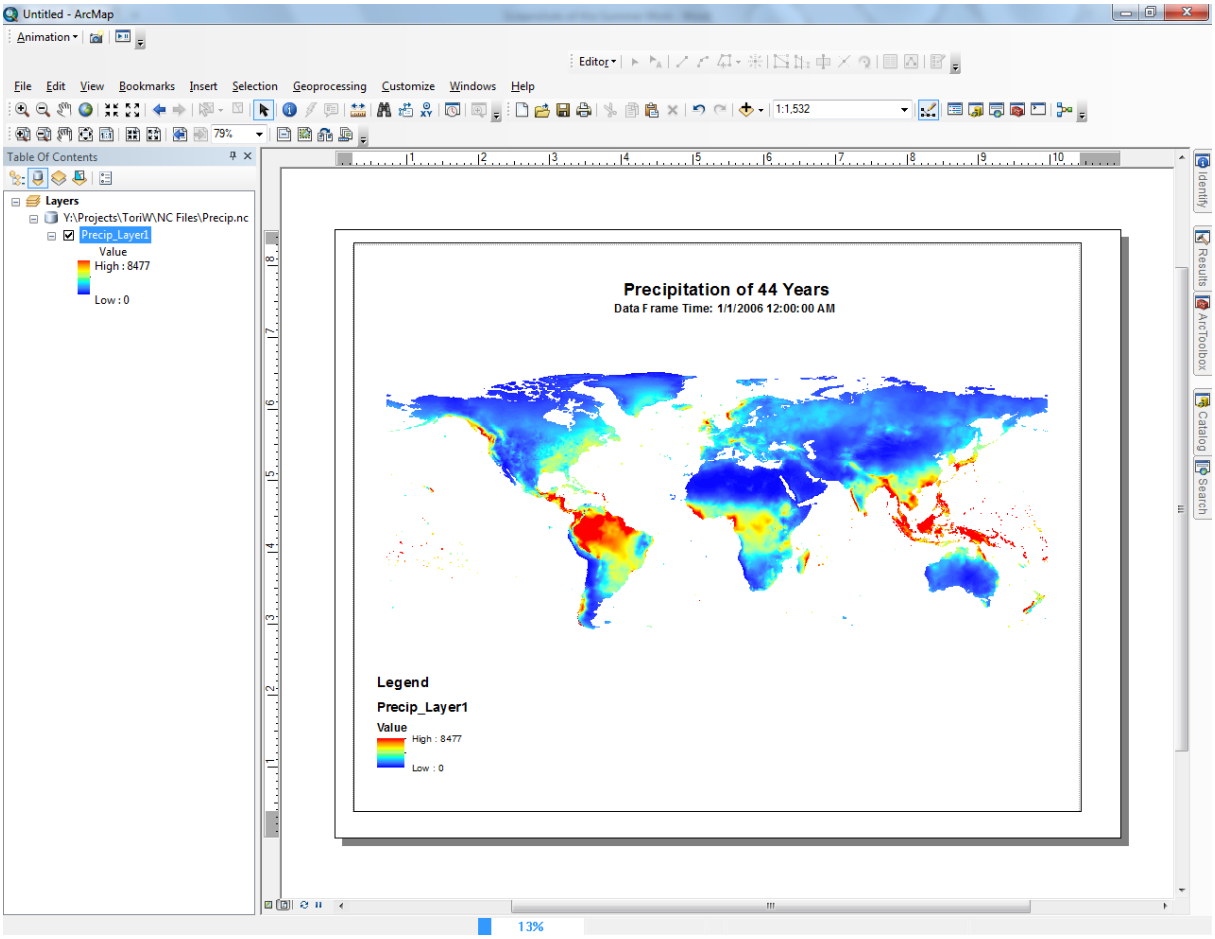
3. Click the Animation button and select Create Time Animation. A message will come up stating it is a success. Next click this button to bring Animation Controls.
    a. To find the animation toolbar go into customize and toolbars.



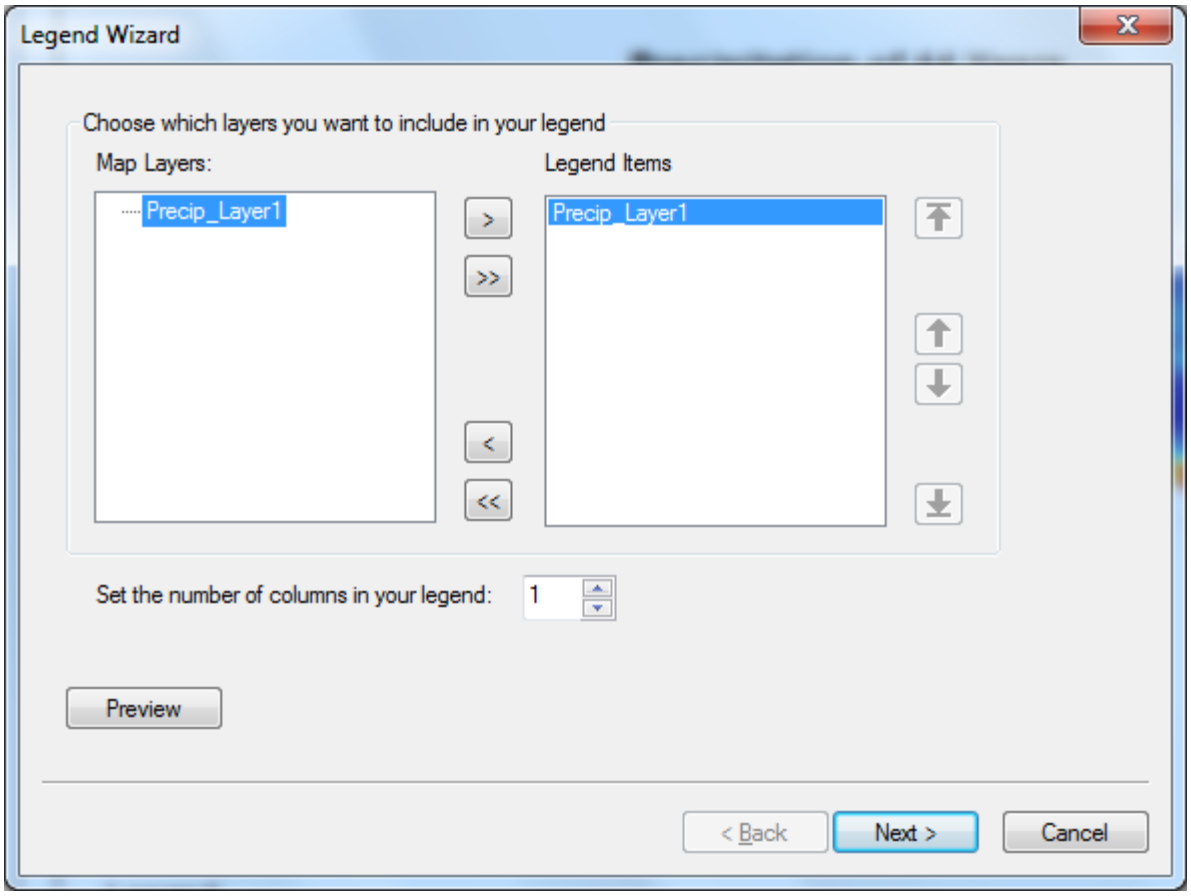
4. Choose the by number of frames and insert how many frames there are. Then check the Play only from and however many frames you have increase the frs. By one. This will allow the last layer/frame to be seen.



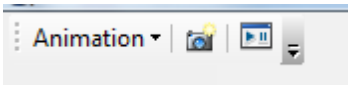
5. To set the Legend to a fix value. Change Type in the Stretch section to Minimum-Maximum. Then check the Edit High/Low Values and edit the high and low under values. Then uncheck the Edit High/Low Values. This will change all the layers to this fixed legend.



6. Then you change to layout, and add the legend, title and data frame time.

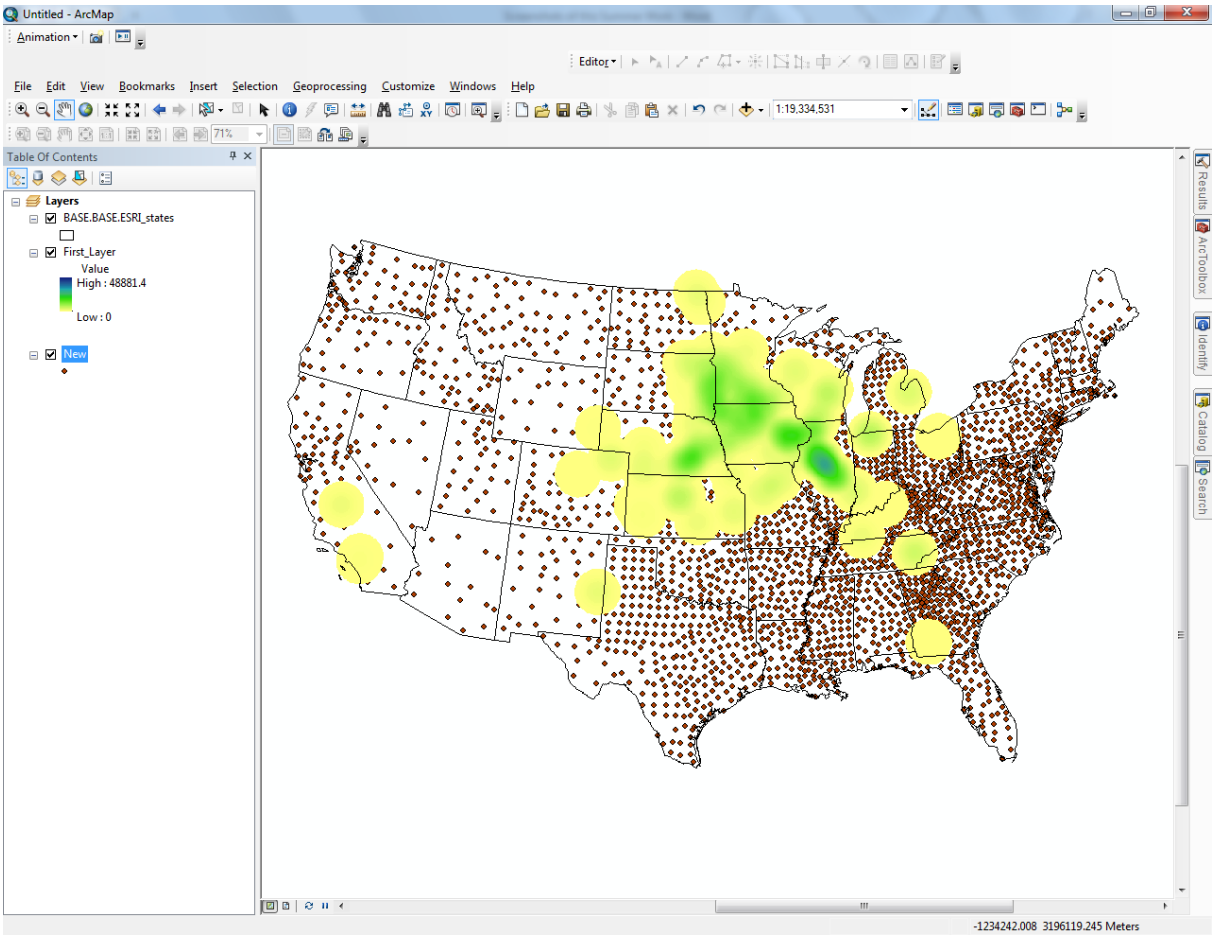


7. To create a legend go to insert and click legend. If there is an item you do not want move it to the left and press next until you get to finish.
8. To create title go to insert and click Title. It will allow you to type in a title.
9. To create a data frame time go to insert, dynamic text and click data frame time.
    a. Data frame time shows what frame the video is show, by showing the date and time.

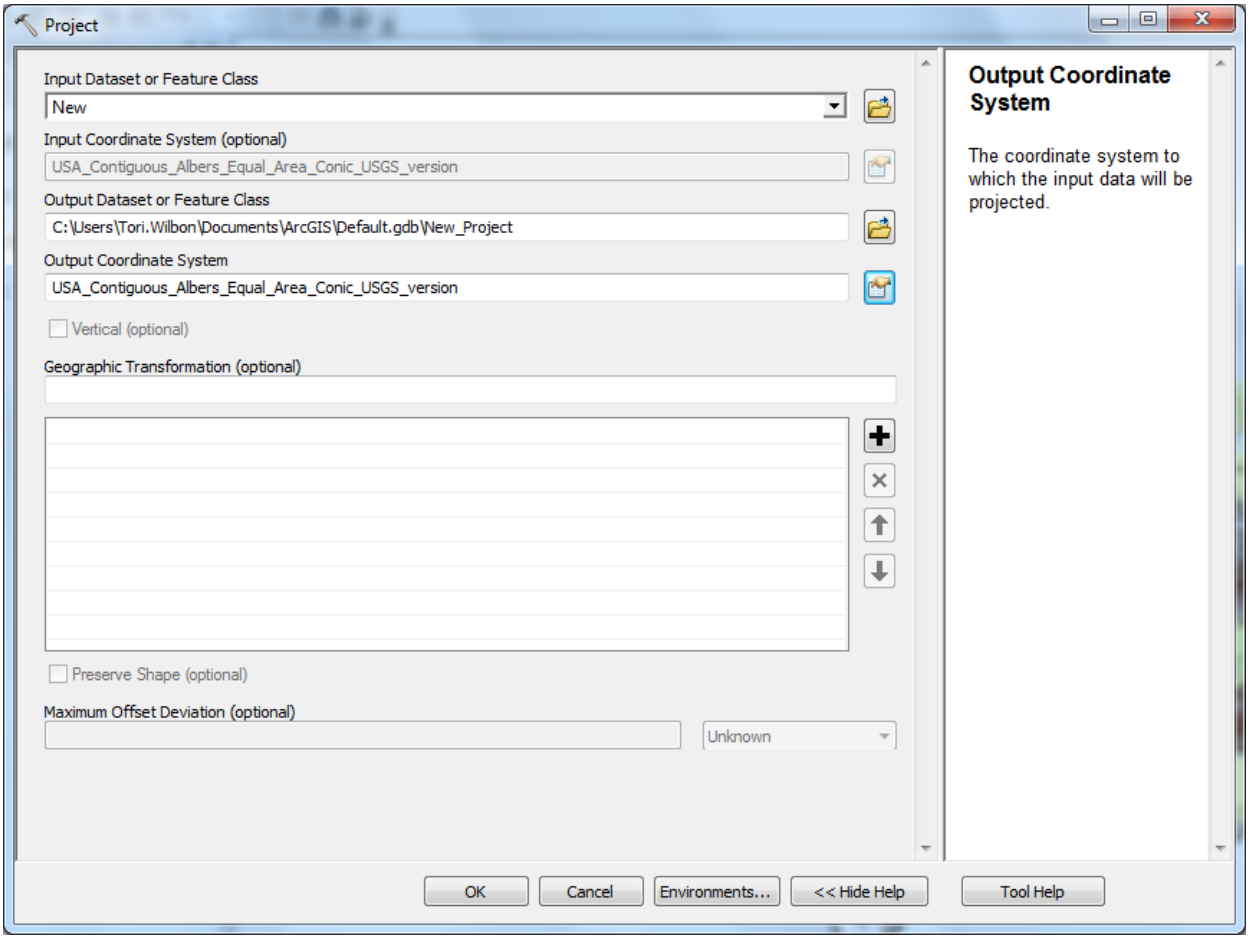


10. Then click the Animation button again and click Export Animation.

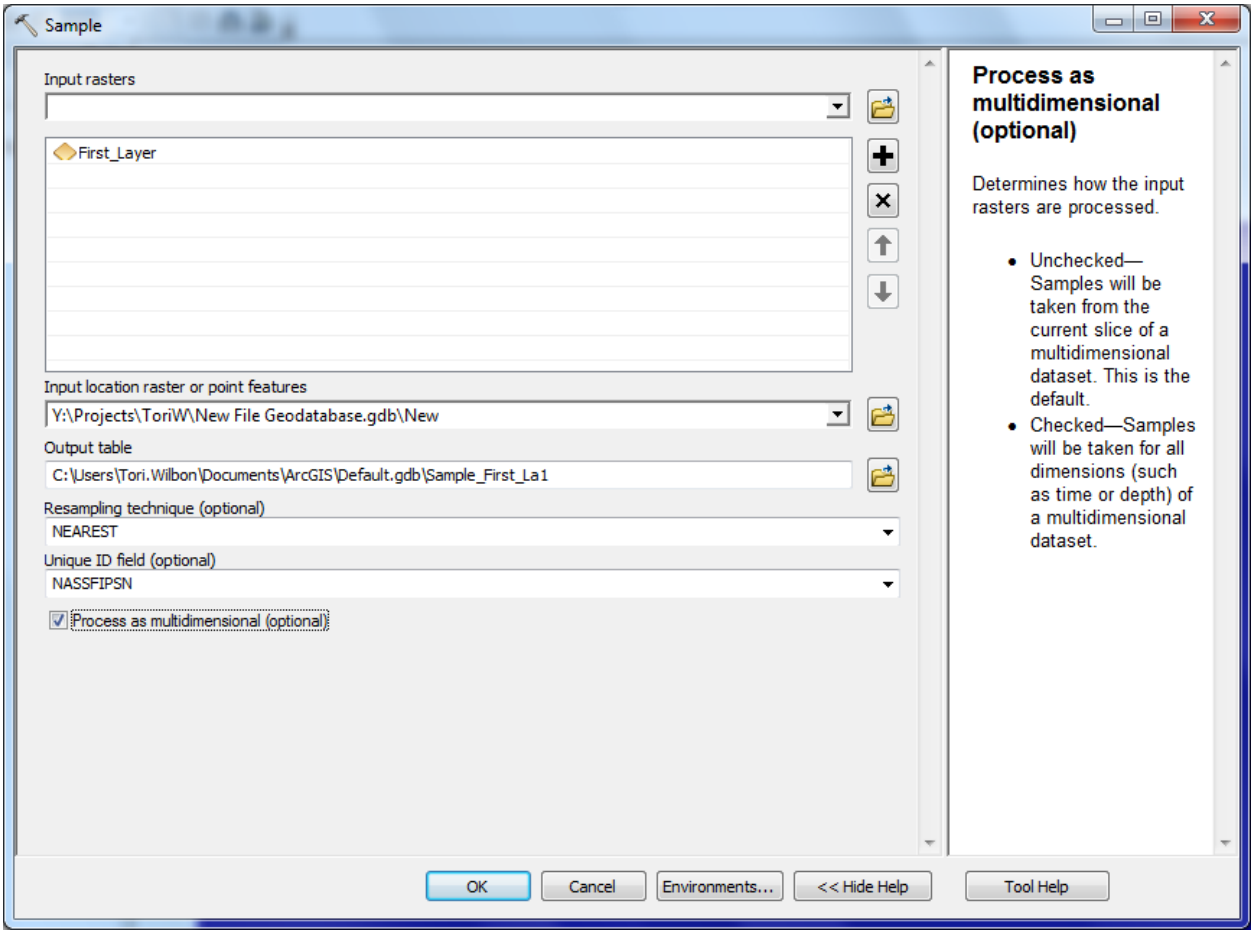
Using the sampling tool on Raster Layer



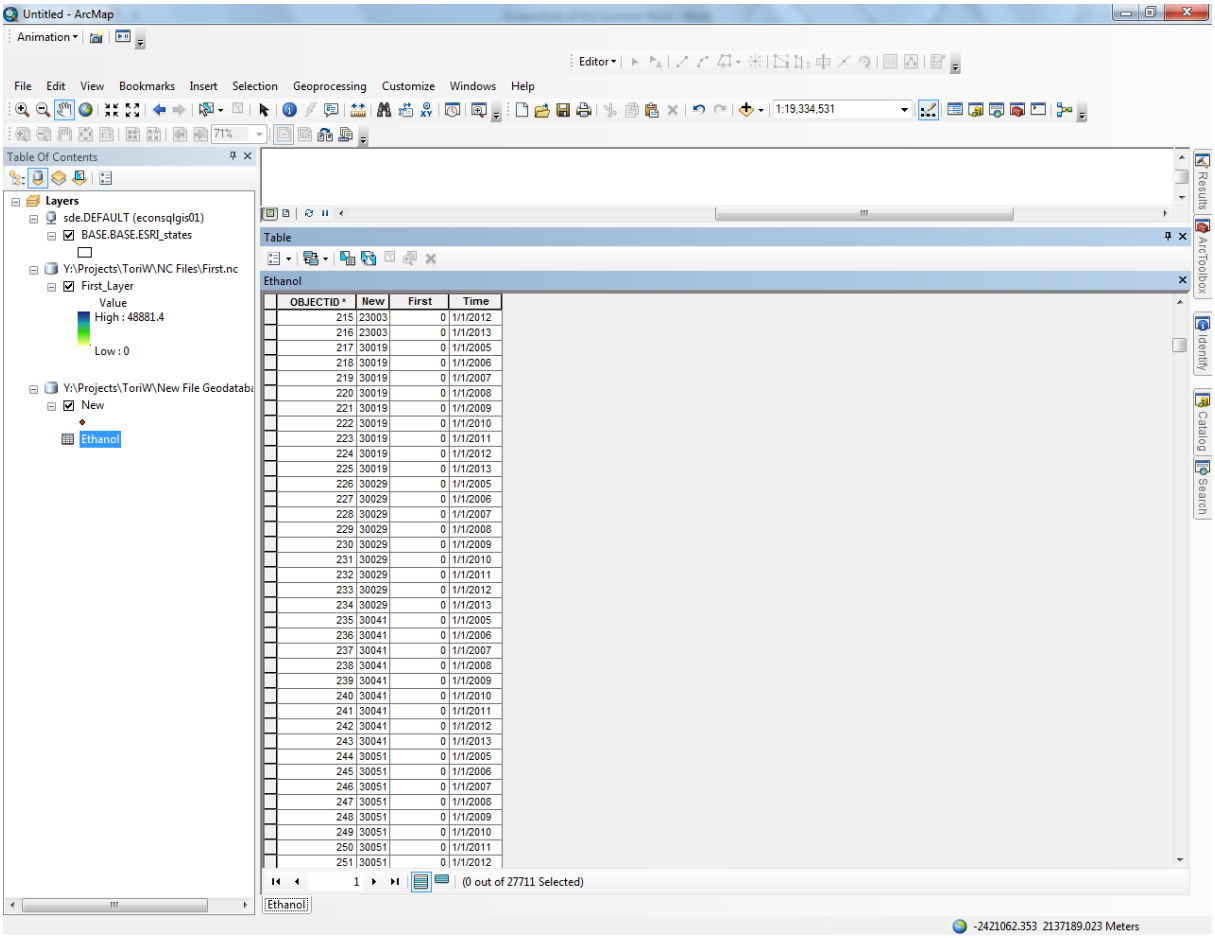
1. Have the layers you want to use in the tool.
    a. I have the Ethanol NetCDF Layer.
    b. The ESRI States Layer
    c. The NASS_Counties_Points Layer



2. Make sure the layers coordinate systems are equal. If not go to Data Management Tool→ Projections and Transformations→Project.
    a. You will not be able to use the Sample tool without the projections equaling.



3. Then you perform the sample tool. Since this raster is from a NetCDF file make sure you check Process as multidimensional. Spatial Analyst Tool→ Extraction→Sample.



4. Results will be in a table

NetCDF using Zonal Statistics

Precip Zonal Rasters.py - Y:\Projects\ToriW\Scripts\Precip Zonal Rasters.py (3.4.4)

File Edit Format Run Options Window Help

```
#Modules needed
import arcpy
import numpy
import netCDF4

total = 0

def Calculate (pre):
      global total
      total = total+pre

def NewFile(total):
      global lat, lon
      for j in lat:
            ji = numpy.where(j==lat)
            for k in lon:
                  ki = numpy.where(k==lon)
                  if (total[ji,ki]>90000):
                        total[ji,ki]=None

      lati = numpy.arange(-89.75,90.25,0.5)
      long = numpy.arange(-179.75,180.25,0.5)

      row = 360
      column = 720

      dataset = netCDF4.Dataset(r'Y://Projects//ToriW//NC Files//JulyRaster.nc', 'w', format = 'NETCDF4_CLASSIC')
      dataset.set_fill_off()
      dataset.createDimension('lats',row)
      dataset.createDimension('lons',column)

      lats = dataset.createVariable('lats', 'f4',('lats'))
      lats.units = 'degree_north'

      lons = dataset.createVariable('lons','f4',('lons'))
      lons.units = 'degree_east'

      lats[:] = lati
      lons[:] = long
      Pre = dataset.createVariable ('Pre',numpy.float64, ('lats','lons'))

      Pre[:,:] = total

      dataset.close()
      Zonal()
def Zonal ():
      v50_states_shp = "Y:\\Data\\Spatial\\Projected\\50_states.shp"
      JulyRaster_nc = "Y:\\Projects\\ToriW\\NC Files\\JulyRaster.nc"
      Holder = "Holder"
      JulyRasters = "Y:\\Projects\\ToriW\\New File Geodatabase.gdb\\JulyRasters"

      # Process: Make NetCDF Raster Layer
      arcpy.MakeNetCDFRasterLayer_md(JulyRaster_nc, "Pre", "lons", "lats", Holder, "", "", "BY_VALUE")

      # Process: Zonal Statistics as Table
      arcpy.gp.ZonalStatisticsAsTable_sa(v50_states_shp, "STATE_FIPS", Holder, JulyRasters, "DATA", "MEAN")
```

```
#Main
f =netCDF4.Dataset(r'Y://Data//Databases//BADC//v3.23//Precip//cru_ts3.23.1971.1980.pre.dat.nc', 'r',format='NetCDF4_CLASSIC')
lon = numpy.array(f.variables['lon'][:])
lat = numpy.array(f.variables['lat'][:])
date =numpy.array(f.variables['time'][:])
precip = numpy.array(f.variables['pre'][:], dtype=type(f.variables))

S= precip
count=0
a = 0
for i in date:
      count =count +1
      if (count ==7):
            timei=numpy.where(i==date)
            pre = S[a,:,:]
            Calculate (pre)
      elif (count ==12):
            count = 0
      a=a+1

NewFile(total)
```

Ln: 4 Col: 14

1. This is the code to take the sum from the NetCDF file. Then to use the raster layer and the zonal statistics to find the mean in each state.
    a. So the libraries needed are arcpy, numpy and netCDF4
    b. I start with setting total to 0 since I know that it will become a global variable later in a function.
    c. Since functions have to be defined before calling it, this is why the three functions Calculate, NewFile and Zonal are above the main function.

d. In the main function I set f equal to the NetCDF that I want to use in the code.

e. Then I extracted the variables lon, lat, time and pre.

f. Then I set S to precip so I am able to manipulate it.

g. The count variable is to tell what month it is since the BADC data starts on January and ends in December. The a variable is to tell what time S should be in when gets to the 7th month.

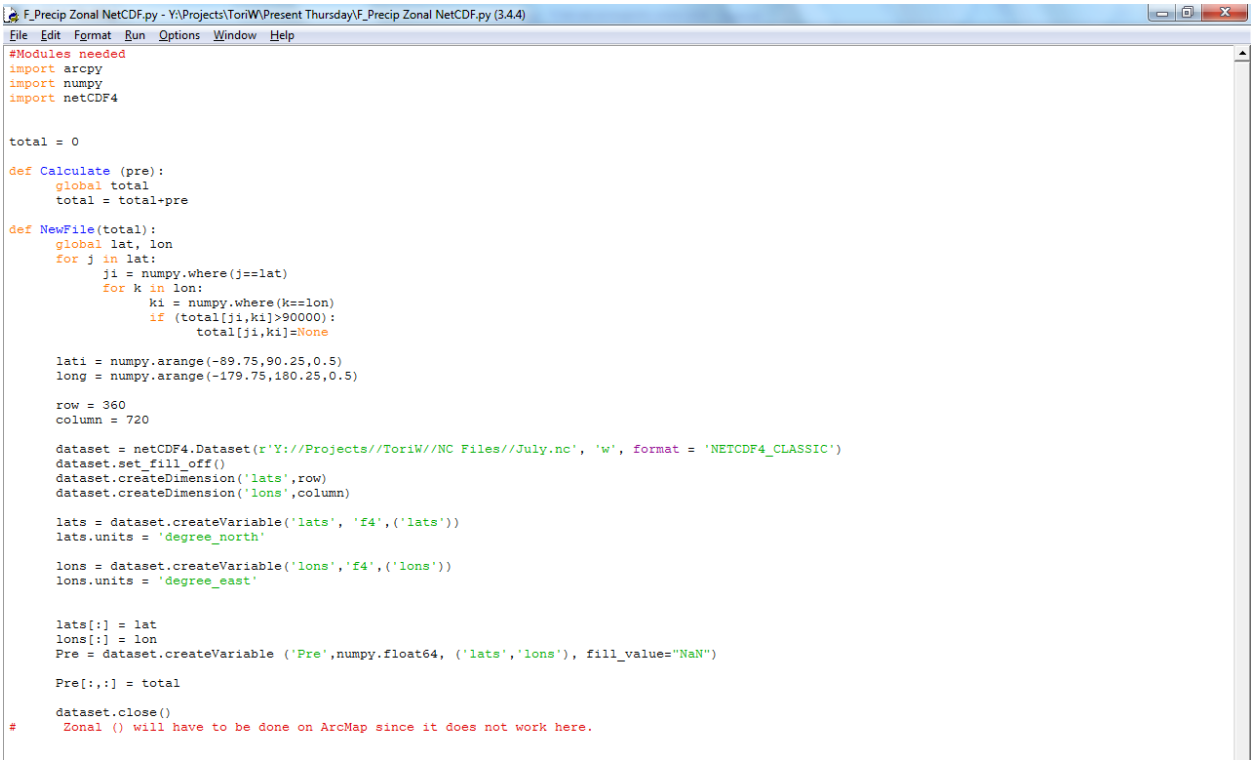
h. On the 7th month the calculate function is called and the value S is sent to the function.

i. On the 12th month the count is changed back to zero and months start over again.

j. After the loop is done then the NewFile is called.

k. In the calculate function I define total as a global variable so I am able add values into it. Then I add itself to pre which is the precipitation of July.
l. In the Newfile function I define lat and lon from the original NetCDF as global variables.
m. Create two for loops to go through the sum data and remove all null values.
n. You set the latitude and longitude extent, by using the numpy.arange. this can either come at this point or you can wait after you create the variables.
o. You can either create the variables for the number of rows and columns or you can type that in into the createDimension.
p. Set dataset to the NetCDF file that is being created, w means write and you can set the format to NETCDF4_CLASSIC.
q. Set the fill off. This prevents the file from opening the NetCDF file and bring the null values back.
r. Create the longitude and latitude dimensions. You name them which mine are name lats and lons and you set the size/
s. Next create the variables longitude and latitude. Make sure you set the first space and last space the same name or else you will not get the correct coordinates. The second space is where the datatype goes, which I chose f4. The units again it doesn't matter if you have it there or not. It does not alter you data.
t. Now you populate the lats and lons variables with the lati and long

u. Create the pre variable by using the two dimensions lats and lons.\
v. Populate the Pre variable make sure you have it looking like Pre[:,:] and it be equal to total.
w. Make sure you close the file. If you do not then the Pre variable values will become null. Then the zonal function is called.
x. In the Zonal function set a value to equal my shape file and one to equal the raster file that needs to be open.
y. Had to create a file that holds the raster layer so this is where holder comes from.
z. The set what the new table from the zonal statistics should equal.
aa. Ran the Make NetCDF Raster Layer and the Zonal Statistics as Table.



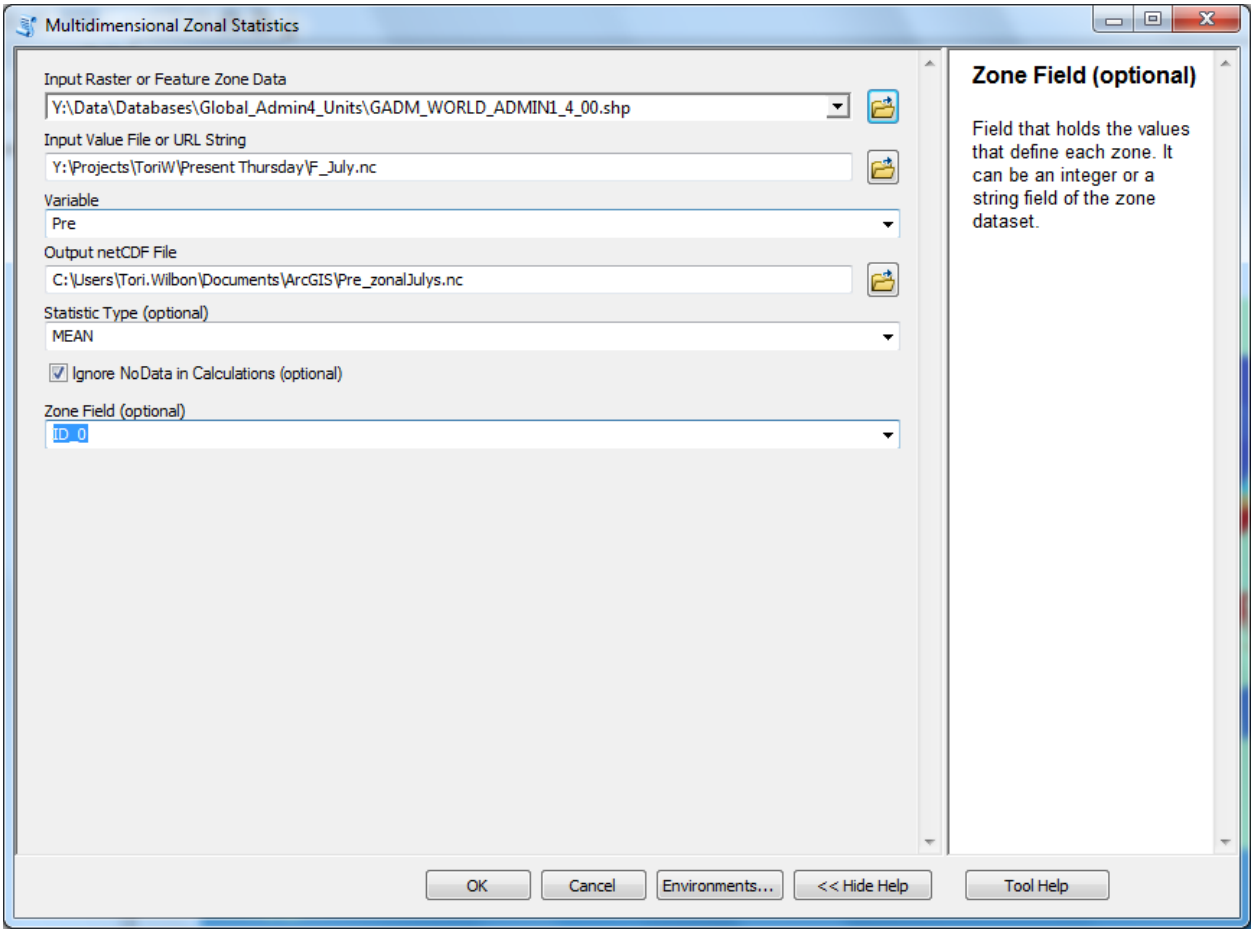

```
#Main
f =netCDF4.Dataset(r'Y://Data//Databases//BADC//v3.23//Precip//cru_ts3.23.1971.1980.pre.dat.nc', 'r',format='NetCDF4_CLASSIC')
lon = numpy.array(f.variables['lon'][:])
lat = numpy.array(f.variables['lat'][:])
date =numpy.array(f.variables['time'][:])
precip = numpy.array(f.variables['pre'][:], dtype=type(f.variables))

S= precip
count=0
a = 0
for i in date:
      count =count +1
      if (count ==4):
            timei=numpy.where(i==date)
            pre = S[a,:,:]
            Calculate (pre)
      elif (count ==12):
            count = 0
      a=a+1

NewFile(total)
```

Ln: 34 Col: 0

2. This is the code for only NetCDF. The difference between the codes are in the new NetCDF file the variable Pre has a fill_value of “NaN”. This is done, because if you do not put that in then the null values will come back into the data, giving wrong data. Also if you put NaN in lon or lat you will get an error when you run the multidimensional zonal statistics tool.



3. Because my computer does not allow me to run the multidimensional zonal statistics tool in python you will have to go into ArcMap and run the tool.