

# Using Jupyter Notebooks to Perform Atmospheric Data Analytics

This notebook demonstrates a spaghetti plot of the surface air temperature (tas) monthly anomaly of two reanalyses, as referenced by:

Potter, G., L. Carriere, J. Hertz, M. Bosilovich, D. Duffy, T. Lee, and D. Williams, 2017: Enabling Reanalysis Research Using the Collaborative Reanalysis Technical Environment (CREATE). Bull. Amer. Meteor. Soc. doi:10.1175/BAMS-D-17-0174.1

To run this notebook you will need to install cdat using conda:

1. Install conda from Anaconda: <https://www.continuum.io/>.
2. Install cdat from UV-CDAT: <https://github.com/UV-CDAT/uvcdat/wiki/Install-using-Anaconda>.
3. Start Jupyter Notebook in a terminal window ("jupyter-notebook").
4. From the Jupyter top menu in the browser window, select "File" and "Open" this notebook.
5. A "Kernel not found" dialog window may appear after opening the notebook, asking you to select a kernel. Choose "Python [conda env:cdat8]" from the pull-down menu and choose "Set Kernel". The current kernel is displayed at the top far right of the notebook page. The kernel can be changed from the menu, select "Kernel" and "Change kernel" for a list of available kernels.
6. Select "Kernel" then "Restart & Run All".

See [https://esgf.nccs.nasa.gov/projects/create-ip/use\\_cases](https://esgf.nccs.nasa.gov/projects/create-ip/use_cases) for complete instructions.

In [1]:

```
# import CDAT utilities to handle the data
import cdms2
import vcs, cdutil
# customize the variable name and the description of the plot
varin='tas'
# give the new variable (Surface Temperature Anomaly) a name
varid='Surface Air Temperature Anomaly'
# set a fixed minimum and maximum for the plot
plotmin=-.6
plotmax=.6
# set units
units='K'
# set the latitude range for the plot
lat1=-90
lat2=90
# more reanalyses:
https://dataserver.nccs.nasa.gov/thredds/catalog/bypass/CREATE-IP/catalog.html
# this example selects data from MERRA2 and ERA-Interim reanalyses
file1='https://dataserver.nccs.nasa.gov/thredds/dodsC/bypass/CREATE-IP/rean
```

```

file1='https://dataserver.nccs.nasa.gov/thredds/dodsC/bypass/CREATE-IP/rean
analysis/MERRA2/mon/atmos/tas.ncml'
file2='https://dataserver.nccs.nasa.gov/thredds/dodsC/bypass/CREATE-IP/rean
analysis/ECMWF/mon/atmos/tas.ncml'
# set end date of the data time period
# the date below is the most recent data in the catalogue for MERRA2 and
ECMWF ERA-I
# dates are monthly mean, reflecting points in mid-month, use '15' for Febr
uary ('1989-02-15')
end_date='2017-12-16'

# open the file from the OPeNDAP address
f1=cdms2.open(file1)
# read the variable (end date can be a date like '1989-01-16')
# dates are monthly mean, reflecting points in mid-month, use '15' for Febr
uary ('1989-02-15')
tas1=f1(varin,lat=(lat1,lat2),time=('1980-01-16',end_date))
t=tas1.getTime()
cdutil.setTimeBoundsMonthly(tas1)
# calculate temperature anomaly
tas1=cdutil.times.ANNUALCYCLE.departures(tas1)
# calculate global average
tas1=cdutil.averager(tas1(squeeze=1),axis='xy')

# do the same for the second reanalysis
f2=cdms2.open(file2)
tas2=f2(varin,lat=(lat1,lat2),time=('1980-01-16',end_date))
cdutil.setTimeBoundsMonthly(tas2)
tas2=cdutil.times.ANNUALCYCLE.departures(tas2)
tas2=cdutil.averager(tas2(squeeze=1),axis="xy")
# reset variable ID
tas1.id=''
tas2.id=''

# initiate plotting package
x=vcs.init()
x.drawlogooff()
gm = x.createId()
# data range in Y
gm.datawc_y1= plotmin
gm.datawc_y2 = plotmax
gm.linewidth = 2
# set colors of lines
colors = ["red","black",]
template = vcs.createtemplate()
# blanks for extra fields at top of canvas default is to show the mean min
and max values
template.blank(["dataname","legend","mean","min","max","source"])
template.scale(.8,'x')
template.moveto(.2,.2)
# locate the legend on the plot
template.legend.x1=.3
template.legend.x2=.4
template.legend.y1=.68
template.legend.y2=.78
# put variable name in center as a title
# template.dataname.x=.45
template2 = vcs.createtemplate(source=template.name)
template2.legend.priority=1
# set colors for legend
# template2.drawLinesAndMarkersLegend(x,

```

```

["red","blue","green","orange","cyan","magenta","black"],
["solid","solid","solid","solid","solid","solid","solid"],[2,2,2,2,2,2,2],
["red","blue","green","orange","cyan","magenta","black"],
['dot','dot','dot','dot','dot','dot','dot'],[0,0,0,0,0,0,0],["CFSR","MERRA",
,"MERRA2","JRA-55","ERA-Interim","20CRv2c","GPCP"])
template2.drawLinesAndMarkersLegend(x,["red","black"], ["solid","solid"],[2
,2], ["red","black"],['dot','dot'],[0,0],["MERRA2","ERA-Interim"])

for i,ts in enumerate([tas1,tas2]):
# make a copy of graphic method to change color
    tmp_gm = x.createId(source=gm.name)
    tmp_gm.linecolor= colors[i]
    tmp_gm.marker=0
    d = x.plot(ts,tmp_gm,template,bg=True)
    # put plot in the Jupyter Notebook
# d
header = x.createtext()
header.To.height = 36
header.To.halign = "center"
header.To.valign = "top"
header.x = .55
header.y = .89

header.string = [varid]

x.plot(header, bg=1)
txt=x.createtext()
txt.string=units
txt.height=20
txt.x=.15
txt.y=.5
txt.angle=-90
x.plot(txt)
# to save plot on your computer, edit path and uncomment the next line
# x.png('~\myplots\plot1.png')
d

```

Out[1]:



