Manipulating Spatio-temporal Environmental Data in R

Using the `ncdf` package

Jon Hobbs

UseR! 2007
August 9
`http://www.public.iastate.edu/~jonhobbs/user`

Outline

Outline

Environmental Data

- Characteristics of environmental datasets, especially in meteorology and oceanography
  - Spatial fields - regular or irregular, possibly 3D
  - Observations across time
  - Multivariate - e.g., temperature, wind, pressure
- Bookkeeping is important
  - Observation Locations
  - Observation Times
  - Measurement Units
- Large datasets

Data Storage/Access

- A "standard" data format could be useful
- Some desirable qualities
  - Self-describing data files
  - Handle space and time in a reasonable way
  - Efficient storage
- Data access should be
  - Fast (relatively)
  - Piecewise, if desired

Example - Data Expo

- 2006 Data Exposition used data derived from NASA satellite observations
- Monthly observations from January, 1995 - December 2000
- Spatial domain is a regular grid of 24×24 locations
- Seven variables
  - Ozone
  - Surface pressure
  - Two temperature measurements
  - Cloud cover at three vertical levels

Data Expo

Data was provided as a single text file for each variable and month

```
VARIABLE : Mean Near-surface air temperature (kelvin)
FILENAME : ISCCPMonthly_avg.nc
FILEPATH : /usr/local/fer_data/data/
SUBSET : 24 by 24 points (LONGITUDE-LATITUDE)
113.8W 111.2W 108.8W 106.2W 103.8W ...
36.2N / 51:  301.4 301.4 301.4 300.5 285.8 ...
33.8N / 50:  301.4 301.4 288.3 287.3 302.8 ...
31.2N / 49:  301.0 301.0 301.0 301.0 301.9 ...
28.8N / 48:  301.0 301.0 301.0 292.7 302.3 ...
26.2N / 47:  301.4 301.9 301.9 301.9 301.4 ...
```

Data Expo

- The data format is nice for looking at spatial fields of individual variables.
- Other combinations take some work
  - Time Series at individual locations
  - Relationships between variables
- A little programming can get the data into different desirable formats.
- Can we get around this?

Outline

NetCDF

- The Network Common Data Form, managed by Unidata, provides an approach to organizing and storing multivariate space-time data.

  `http://www.unidata.ucar.edu/software/netcdf`

- From Unidata:

  NetCDF is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.

NetCDF Fundamentals

- The NetCDF core is a set of C and Fortran libraries, which are prerequisites for higher-level interfaces.
- NetCDF data files are platform-independent binary files.
- A data file contains a header, or metadata, that describes the file contents.
- Extension is usually ".nc"

NetCDF Fundamentals

- A NetCDF file has some key components
- Dimensions
  - Reference spatial dimensions and time
  - Each dimension has a specified length
  - One dimension can have "unlimited" length
  - Data Expo dimensions are X (east-west), Y (north-south) and time
- Attributes
  - Strings describing measurement units, long names, or observation times
  - Numerical values giving valid variable minima and maxima

NetCDF Components

- Variables
  - Each variable has a specific ordering of dimensions defining how data is stored and accessed
  - Each variable has a data type (float, integer, character, etc.)
  - Data Expo NetCDF file has 10 variables - satellite variables plus elevation, latitude, longitude
  - Elevation is a float (single precision) with dimensions (X,Y)

Outline

The `ncdf` Package

- At last check, three contributed R packages utilize NetCDF
  - `ncdf`
  - `ncvar`
  - `RNetCDF`
- All three require installation of the Unidata NetCDF libraries first.
- `ncvar` requires `RNetCDF`

The ncdf Package

- The ncdf provides high-level read/write capability for NetCDF files in R.
- Written by David Pierce

  http://cirrus.ucsd.edu/~pierce/ncdf
- Installation
  - Mac/Linux: Define path to NetCDF libraries/includes
  - Windows: Copy NetCDF dlls to ncdf library directory
- ncdf objects are returned with calls to open.ncdf or create.ncdf

Working with ncdf

```
> library(ncdf)
> nc1 = open.ncdf("expo.nc")
> print(nc1)
"file expo.nc has 3 dimensions:"
"X Size:  24"
"Y Size:  24"
"Month Size:  72"
"-------------------------"
"file expo.nc has 10 variables:"
"float cloudhigh[X,Y,Month]"
...
"float temperature[X,Y,Month]"
"float elevation[X,Y]"
"float latitude[Y]"
"float longitude[X]"
```

Working with `ncdf`

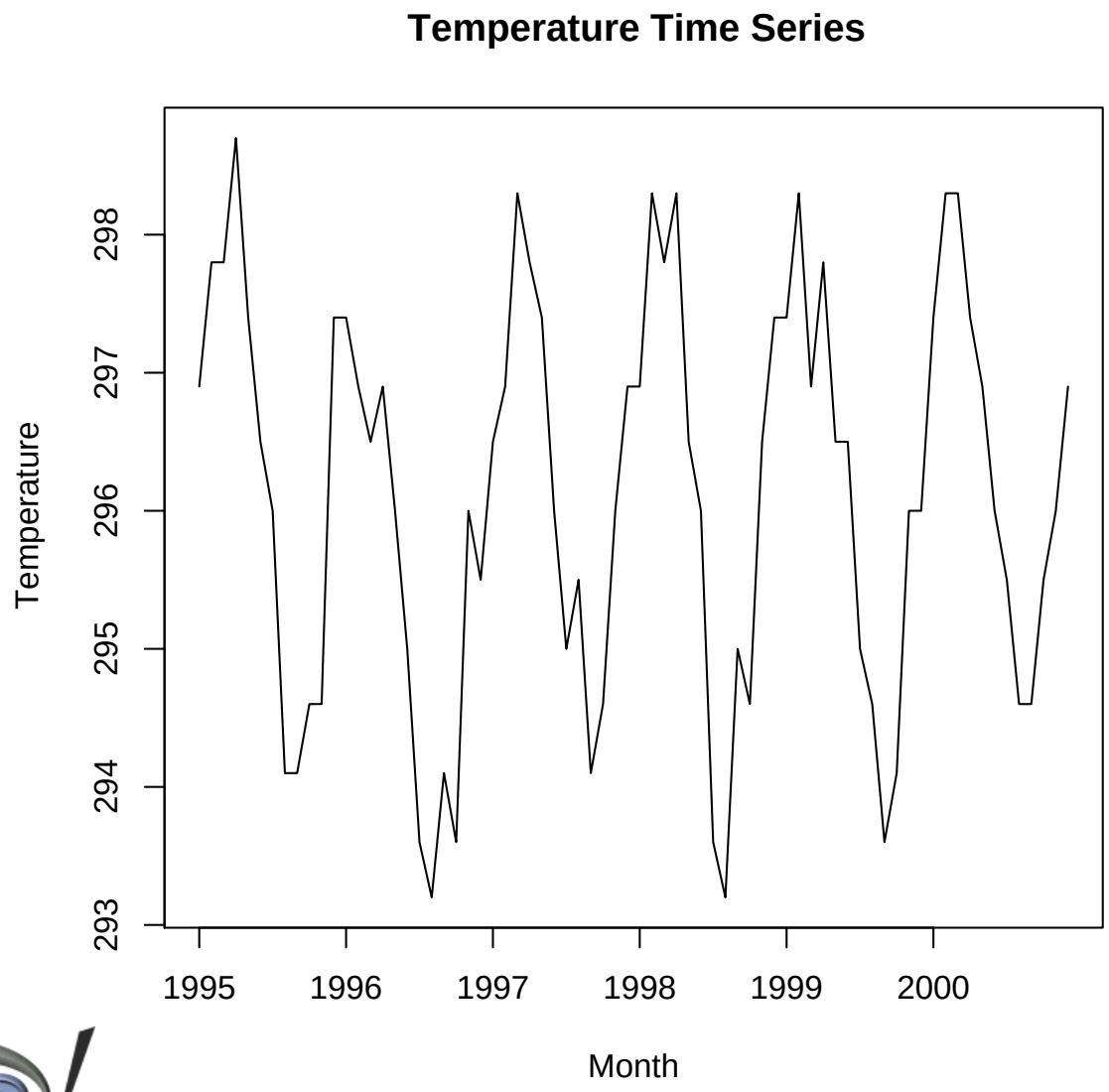
- Functions in `ncdf` are combinations of NetCDF components and actions
- Components
  - Dimensions - `dim`
  - Attributes - `att`
  - Variables - `var`
- Actions
  - Define - `def`
  - Read - `get`
  - Write - `put`
- `dim.def.ncdf` creates a new dimension
- `get.var.ncdf` reads a variable into an R array

Data Expo

```
> oz = get.var.ncdf(nc1,"ozone")
> dim(oz)
[1] 24 24 72
> lat = get.var.ncdf(nc1,"latitude")
> dim(lat)
[1] 24
> lon = get.var.ncdf(nc1,"longitude")
> dim(lon)
[1] 24

> tmpset =
get.var.ncdf(nc1,"temperature",start=c(1,1,1),count=c(1,1,72))
> dim(tmpset)
[1] 72
```

Data Expo


Temperature Time Series
Temperature
293
294
295
296
297
298
1995
1996
1997
1998
1999
2000
Month

Data Expo

Some code to plot spatio-temporal ozone data

```
library(RColorBrewer)
brk = seq(220,400,by=20)
lvec = c(1:72,rep(73,12))
layout(matrix(lvec,nrow=7,byrow=TRUE),
    heights=c(rep(1,6),0.5),widths=c(rep(1,12)))
par(mai=c(0.05,0.05,0.05,0.05))
for (i in 1:72) {
   image(lon,lat,z=oz[,,i], col=brewer.pal(9,"YlOrRd"),
     axes=F,pty="s",ylab="",xlab="",breaks=brk)
   map("world",add=TRUE)
   abline(h=0)
}
```

Data Expo

Adding a legend

```
par(mai=c(0,1,0,1))
plot(20,1,xlim=c(0,20),ylim=c(0,1),axes=FALSE,type="n",
    xaxt="n",yaxt="n",xlab="",ylab="",frame.plot=FALSE)
xl = seq(7.75,11.75,by=0.5)
yb = rep(0.5,9)
xr = seq(8.25,12.25,by=0.5)
yt = rep(0.8,9)
rect(xl,yb,xr,yt,col=brewer.pal(9,"YlOrRd"))
text(seq(7.75,12.25,by=0.5),rep(0.3,10),
labels=paste(brk),cex=0.75)
```

Clean Up

- Close a NetCDF file with `close.ncdf(nc1)`
- The data arrays can be saved in the R workspace
- Watch out for large arrays that may have been created

Outline

Discussion

- Irregular spatial data can be handled by NetCDF, likely just one spatial dimension.
- Another data format, Gridded Binary (GRIB), is often used in meteorology, but no R package yet.
- NetCDF and GRIB work well when the data collection scheme remains consistent
  - Ideal for computer model output
  - What to do when observation locations are added or removed, i.e. ragged data?

Questions

- Questions?