Zoo955 - Raster Data

Hilary Dugan

February 20, 2018

What is a Raster? (from raster package)

A raster is a spatial (geographic) data structure that divides a region into rectangles called 'cells' (or 'pixels')

- Each cell has 1 or more data values

Such a data structure is also referred to as a 'grid' and is often contrasted with 'vector' data that is used to represent points, lines, and polygons.

Raster advantages & disadvantages over vector data

- Raster data usually requires more storage space than vector data
- Raster data can be more difficult to re-project and scale
- Raster data can be more difficult to update
- Raster data can be computationally less expensive to render graphically
- Easy to plot

Types of Raster Data

- Climate data
- Land cover data
- Elevation data

Many of these are satellite derived, and the resolution depends on the satellite resolution

How is raster data stored?

There are many ways raster data is stored. Each file type is specialized for its use.

Table 1: Some common raster file types

Type	Used for/by	Format
ARC Digitized Raster Graphics (ADRG)	National Geospatial Intelligence Agency	.img or .ovr
JPEG2000	Compression technique for large imagery	.jp2, .j2c, .j2k, or .jpx
GeoTiff	Georeferencing within a TIFF file	.tif, .tiff, or .tff
NetCDF Network Common Data Form	climatology meteorology oceanography	.nc

You don't need to know the file type details, but you will become familiar with with formats can be read into R

Spatial resolution

Each pixel has the same dimensions and shape.

In the case of rasters representing Earth's surface, the size of the area on the surface that each pixel covers is known as the **spatial resolution**

The resolution is tyipcally in meters, but can check the CRS `res(x)` Gives resolution

Using raster data in R

The R Raster package: https://cran.r-project.org/web/packages/raster/vignettes/Raster.pdf

"A notable feature of the raster package is that it can work with raster datasets that are stored on disk and are too large to be loaded into memory (RAM). The package can work with large files because the objects it creates from these files only contain information about the structure of the data, such as the number of rows and columns, the spatial extent, and the filename, but it does not attempt to read all the cell values in memory. In computations with these objects, data is processed in chunks."

Data structures

Raster Layer

A RasterLayer object represents single-layer (variable) raster data

Raster Stack vs. Brick

- There are two classes for multi-layer data the **RasterStack** and the **RasterBrick**
- **RasterStack** can be formed from separate files and/or from a few layers ('bands')
  - RasterStack is a collection of RasterLayer objects with the same spatial extent and resolution
  - Basically a list of RasterLayer objects
- **RasterBrick** can only be linked to a single (multi-layer) file
  - A multilayered object, but can only refer to a single file
  - More effecient processing

Import data

The raster package makes life easy for importing raster data. I have provided two different storage formats for Lake Mendota NLCD data.

1) .grd = native grid format for raster package
2) .tif = geoTIFF

```
library(raster,verbose = F)
nlcd = raster('Data/MendotaNLCD.tif')
nlcd = raster('Data/MendotaNLCD.grd')
nlcd
```

```
## class       : RasterLayer
## dimensions  : 1222, 1073, 1311206  (nrow, ncol, ncell)
## resolution  : 30, 30  (x, y)
## extent      : 517545, 549735, 2246115, 2282775  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0 +ellps=GRS80 +tow
## data source : C:\Users\hdugan\Documents\Rpackages\Zoo955\Lecture4_RasterData\Data\MendotaNLCD.grd
## names       : nlcd_2011_landcover_2011_edition_2014_10_10
```

```
## values      : 11, 95  (min, max)
## attributes  :
##         ID      COUNT Red Green Blue NLCD.2011.Land.Cover.Class Opacity
##  from:   0 7854240512   0     0    0               Unclassified     255
##  to  : 255          0   0     0    0                                  0
```

What is the resolution and extent?

```
res(nlcd)
```

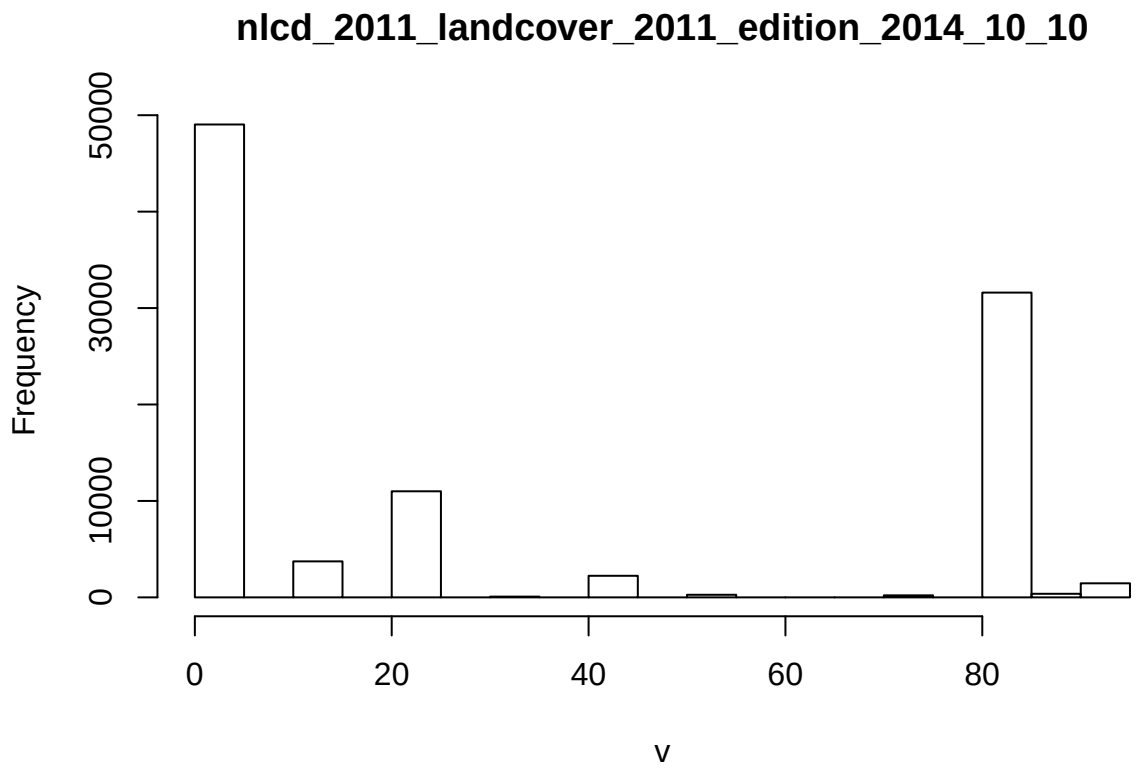
```
## [1] 30 30
```

```
extent(nlcd)
```

```
## class       : Extent
## xmin        : 517545
## xmax        : 549735
## ymin        : 2246115
## ymax        : 2282775
```

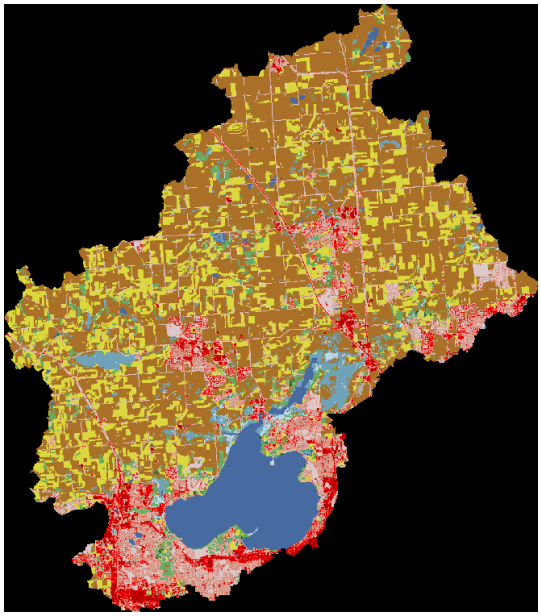
Can explore the data values

```
hist(nlcd)
```

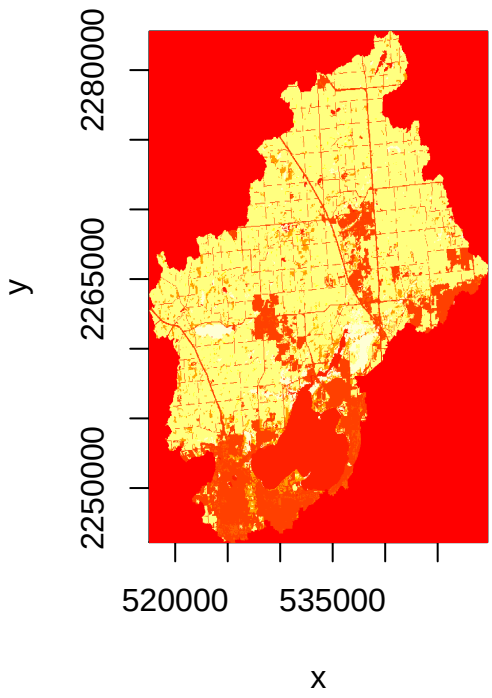


Can plot the data

```
plot(nlcd)
```



```
image(nlcd)
```



Look at the file structure

```
str(nlcd)
```

```
## Formal class 'RasterLayer' [package "raster"] with 12 slots
##   ..@ file    :Formal class '.RasterFile' [package "raster"] with 13 slots
##   .. .. ..@ name        : chr "C:\\Users\\hdugan\\Documents\\Rpackages\\Zoo955\\Lecture4_RasterData\\
##   .. .. ..@ datanotation: chr "INT1U"
##   .. .. ..@ byteorder   : Named chr "little"
##   .. .. .. ..- attr(*, "names")= chr "value"
##   .. .. ..@ nodatavalue : num NA
##   .. .. ..@ NAchanged   : logi FALSE
##   .. .. ..@ nbands      : int 1
##   .. .. ..@ bandorder   : Named chr "BIL"
##   .. .. .. ..- attr(*, "names")= chr "value"
##   .. .. ..@ offset      : int 0
##   .. .. ..@ toptobottom : logi TRUE
##   .. .. ..@ blockrows   : int 0
##   .. .. ..@ blockcols   : int 0
##   .. .. ..@ driver      : chr "raster"
##   .. .. ..@ open        : logi FALSE
##   ..@ data    :Formal class '.SingleLayerData' [package "raster"] with 13 slots
##   .. .. ..@ values    : logi(0)
##   .. .. ..@ offset    : num 0
##   .. .. ..@ gain      : num 1
##   .. .. ..@ inmemory  : logi FALSE
##   .. .. ..@ fromdisk  : logi TRUE
```

```
##   .. .. ..@ isfactor  : logi TRUE
##   .. .. ..@ attributes:List of 1
##   .. .. .. ..$ :'data.frame':    256 obs. of  7 variables:
##   .. .. .. .. ..$ ID                       : int [1:256] 0 1 2 3 4 5 6 7 8 9 ...
##   .. .. .. .. ..$ COUNT                    : num [1:256] 7.85e+09 0.00 0.00 0.00 0.00 ...
##   .. .. .. .. ..$ Red                      : int [1:256] 0 0 0 0 0 0 0 0 0 0 ...
##   .. .. .. .. ..$ Green                    : int [1:256] 0 249 0 0 0 0 0 0 0 0 ...
##   .. .. .. .. ..$ Blue                     : int [1:256] 0 0 0 0 0 0 0 0 0 0 ...
##   .. .. .. .. ..$ NLCD.2011.Land.Cover.Class: Factor w/ 18 levels "","Barren Land",..: 17 1 1 1 1 1 1
##   .. .. .. .. ..$ Opacity                  : int [1:256] 255 255 255 255 255 255 255 255 255 255 ..
##   .. .. ..@ haveminmax: logi TRUE
##   .. .. ..@ min       : num 11
##   .. .. ..@ max       : num 95
##   .. .. ..@ band      : int 1
##   .. .. ..@ unit      : chr ""
##   .. .. ..@ names     : chr "nlcd_2011_landcover_2011_edition_2014_10_10"
##   ..@ legend  :Formal class '.RasterLegend' [package "raster"] with 5 slots
##   .. .. ..@ type      : chr(0)
##   .. .. ..@ values    : logi(0)
##   .. .. ..@ color     : logi(0)
##   .. .. ..@ names     : logi(0)
##   .. .. ..@ colortable: chr [1:256] "#000000" "#00F900" "#000000" "#000000" ...
##   ..@ title   : chr(0)
##   ..@ extent  :Formal class 'Extent' [package "raster"] with 4 slots
##   .. .. ..@ xmin: num 517545
##   .. .. ..@ xmax: num 549735
##   .. .. ..@ ymin: num 2246115
##   .. .. ..@ ymax: num 2282775
##   ..@ rotated : logi FALSE
##   ..@ rotation:Formal class '.Rotation' [package "raster"] with 2 slots
##   .. .. ..@ geotrans: num(0)
##   .. .. ..@ transfun:function ()
##   ..@ ncols   : int 1073
##   ..@ nrows   : int 1222
##   ..@ crs     :Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96 +x_0=0 +y_0=0 +ellp
##   ..@ history : list()
##   ..@ z       : list()
```

Notice there are no data store in the `@data`

```
nlcd@data@values
```

```
## logical(0)
```

```
inMemory(nlcd)
```

```
## [1] FALSE
```

Can use the `getValues` function. Reads off of disk or memory. `getValues(nlcd)`

Notice that there are attributes and a legend

```
atts = nlcd@data@attributes[[1]]
head(atts,20)
```

```
##    ID      COUNT Red Green Blue NLCD.2011.Land.Cover.Class Opacity
## 1   0 7854240512   0     0    0               Unclassified     255
```

```
## 2   1          0   0   249    0                                255
## 3   2          0   0     0    0                                255
## 4   3          0   0     0    0                                255
## 5   4          0   0     0    0                                255
## 6   5          0   0     0    0                                255
## 7   6          0   0     0    0                                255
## 8   7          0   0     0    0                                255
## 9   8          0   0     0    0                                255
## 10  9          0   0     0    0                                255
## 11 10          0   0     0    0                                255
## 12 11  469012527  71   107  160                Open Water     255
## 13 12    1599206 209   221  249         Perennial Snow/Ice     255
## 14 13          0   0     0    0                                255
## 15 14          0   0     0    0                                255
## 16 15          0   0     0    0                                255
## 17 16          0   0     0    0                                255
## 18 17          0   0     0    0                                255
## 19 18          0   0     0    0                                255
## 20 19          0   0     0    0                                255
```

```
nlcd@legend
```

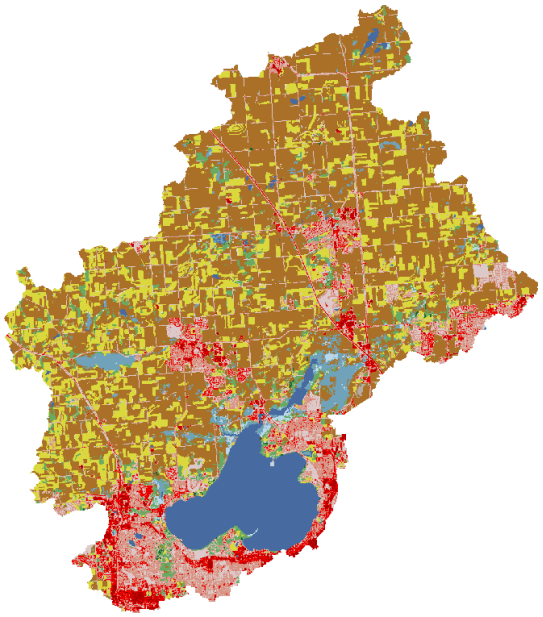
```
## An object of class ".RasterLegend"
## Slot "type":
## character(0)
##
## Slot "values":
## logical(0)
##
## Slot "color":
## logical(0)
##
## Slot "names":
## logical(0)
##
## Slot "colortable":
##   [1] "#000000" "#00F900" "#000000" "#000000" "#000000" "#000000" "#000000"
##   [8] "#000000" "#000000" "#000000" "#000000" "#476BA0" "#D1DDF9" "#000000"
##  [15] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
##  [22] "#DDC9C9" "#D89382" "#ED0000" "#AA0000" "#000000" "#000000" "#000000"
##  [29] "#000000" "#000000" "#000000" "#B2ADA3" "#F9F9F9" "#000000" "#000000"
##  [36] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#68AA63"
##  [43] "#1C6330" "#B5C98E" "#000000" "#000000" "#000000" "#000000" "#000000"
##  [50] "#000000" "#000000" "#A58C30" "#CCBA7C" "#000000" "#000000" "#000000"
##  [57] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
##  [64] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
##  [71] "#000000" "#E2E2C1" "#C9C977" "#99C147" "#77AD93" "#000000" "#000000"
##  [78] "#000000" "#000000" "#000000" "#000000" "#DBD83D" "#AA7028" "#000000"
##  [85] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#BAD8EA"
##  [92] "#B5D3E5" "#B5D3E5" "#B5D3E5" "#B5D3E5" "#70A3BA" "#000000" "#000000"
##  [99] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [106] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [113] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [120] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [127] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
```

```
## [134] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [141] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [148] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [155] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [162] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [169] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [176] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [183] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [190] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [197] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [204] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [211] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [218] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [225] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [232] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [239] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [246] "#000000" "#000000" "#000000" "#000000" "#000000" "#000000" "#000000"
## [253] "#000000" "#000000" "#000000" "#000000"
```

Alter colors

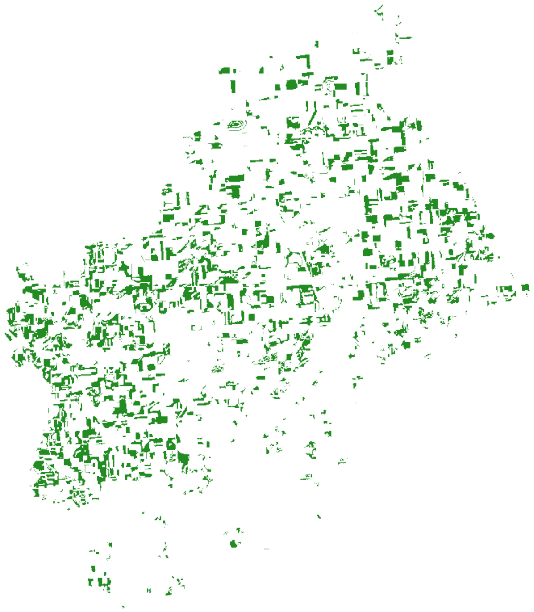
All of the NA data is stored as zeros. In the color legend this is black (#000000). Could change to white.

```
nlcd@legend@colortable[1] = '#ffffff'
plot(nlcd)
```



Could just plot agriculture

```
ag.nlcd = nlcd
ag.nlcd@legend@colortable = '#ffffff'
ag.nlcd@legend@colortable[81] = '#228B22'
ag.nlcd@legend@colortable[82] = '#228B22'
plot(ag.nlcd)
```



Data Counts

```
library(dplyr)
nlcdVals = freq(nlcd)
# Sum values, minus zero values
total = as.data.frame(nlcdVals) %>% dplyr::filter(value >= 1) %>%
  summarise_at('count',sum)

# Total impervious counts
impervious = as.data.frame(nlcdVals) %>% dplyr::filter(value >= 21 & value <= 24) %>%
  summarise_at('count',sum)

# Total ag counts
ag = as.data.frame(nlcdVals) %>% dplyr::filter(value >= 81 & value <= 82) %>%
  summarise_at('count',sum)

perImp = impervious/total
perImp
```

```
##       count
## 1 0.2176047
```

```
perAg = ag/total
perAg
```

```
##       count
## 1 0.6191429
```

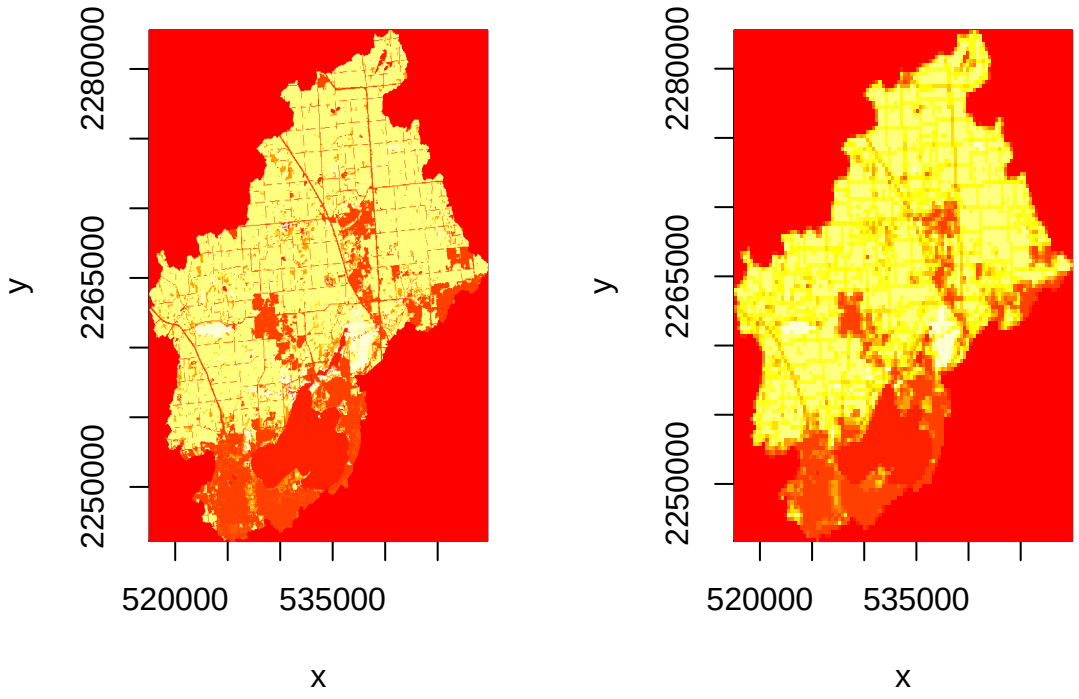
Aggregate data

Can aggregate the data to a lower resolution

```
#aggregate from 30x30 to 300*300 (factor of 10)
nlcd10 <- aggregate(nlcd, fact=10, fun = mean)
res(nlcd10)
```

```
## [1] 300 300
```

```
par(mfrow=c(1,2),mar=c(2,2,1,1))
image(nlcd)
image(nlcd10)
```



Homework

1) What is the percent of forest in the Mendota catchment?

2) What is the area of forest in the Mendota catchment (report in km2)?