Lee McCoy Basic Spatial Reference Card

Packages

```
install.packages("pa1","pa2")
```

Detach package (useful if conflicting commands)

```
detach(“package:name”)
```

Read package from alternative location

```
library(package, lib.loc=”location”)
```

Force R to use the function from a certain package

```
package::function("function")
```

Read/Write Data

Read data

```
read.table(, sep=”\t”)
read.csv()
read.csv(file.choose())
```

Date/Time

“1969-12-31 16:00:00 PST”

For date only:

```
as.Date(data$date, format=’%m%d%Y’)
```

use weekdays(), months(), quarters()

POSIXct – a numerical value in seconds

```
as.POSIXct(data$date, format=’%m%d%Y %H:%M’)
```

POSIXlt – a list of time values (i.e. data$time$day = “Monday”)

```
as.POSIXlt(data$date)
```

`data$date$min` gives minutes

also sec, min, hour, mday, mon, year, wday, yday

*many lists start from 0 instead of 1 (i.e. mon=0-11)

Create a sequence of dates

```
seq(as.Date(“1974-06-17”), by=”days”, length=10)
```

Spatial raster, rgdal

Important spatial packages

```
library(rgdal)
library(raster)
library(sp)
library(maptools)
```

Vector Input/Output

Read in a vector file (i.e. shapefile)

```
readOGR(dsn=”path”, layer=”filename”)
```

Write out a vectore file (i.e. shapefile, KML, GPX)

```
writeOGR(spobject, dsn=”path”,
layer=”filename”, driver=”ESRI Shapefile”)
```

Define the projection of a vector

```
proj4string(data) <- CRS(”…”)
```

Change the projection of a vector

```
spTransform(spobject, crs=”…”)
coordnames(spobject)<-c(“East”,”North”)
```

Other useful things for vectors

`ogrDrivers()` to see available drivers

Name lengths must be ≤10, use `stringr` and

```
str_length(names(data@data))
```

Raster Input/Output

Read in a raster file

```
raster(“path\\rasterfile”)
```

Read in a raster stack (multilayer image)

```
stack(“rasterfile”)
```

Write out a raster file

```
writeRaster(raster, “rasterfile”,
datatype=”FLT4S”, format=”EHdr”)
```

Define the projection of a raster

```
projection(raster) <- CRS(”…”)
```

Change the projection of a raster

```
projectRaster(raster, crs=”…”)
```

** almost all raster functions have an option to give a filename to write to disk instead of memory*

Spatial from Dataframe

Create a spatial object from a dataframe with spatial coordinates.

Coordinates

```
coordinates(utmdf) <- ~east+north
coordinates(latlongdf) <- ~long+lat
```

`coordinates(layer)`see a layers coordinates

Projection

```
proj4string(utmdf)<-CRS(“+proj=utm +zone=10
+datum=NAD83”)
proj4string(latlongdf)<- CRS("+init=epsg:4326")
```

`projection(layer)`see a layers projection

Raster Tools

Cropping a raster with a polygon

```
rasterize(polygon,raster, mask=TRUE)
```

Extract raster values at points

```
value <- extract(raster, point)
```

Spatial points & data frame to spatial points data frame

```
spdf <- SpatialPointsDataFrame(sp,df)
```

Set Extents to Overlap Area

```
intersectExtent(raster1, raster2)
```

Vector Tools

Create random points in a polygon

```
points <- spsample(poly,#pts, type="random")
```

Extract polygon values at points

```
temp <- over(point, poly)
```

Polygon From Extent

```
polygonFromExtent(x, sp=TRUE)
```

Database Access

```
library(RODBC)
con<- odbcConnectAccess2007(“Database.accdb”)
  data<- sqlQuery(con,”SELECT * FROM Table”)
close(con)
```

Save data back to database (be careful!)

```
sqlSave(con, temp, tablename="TABLE", append-T,
rownames=F)
```