GISDAY: R AND GIS

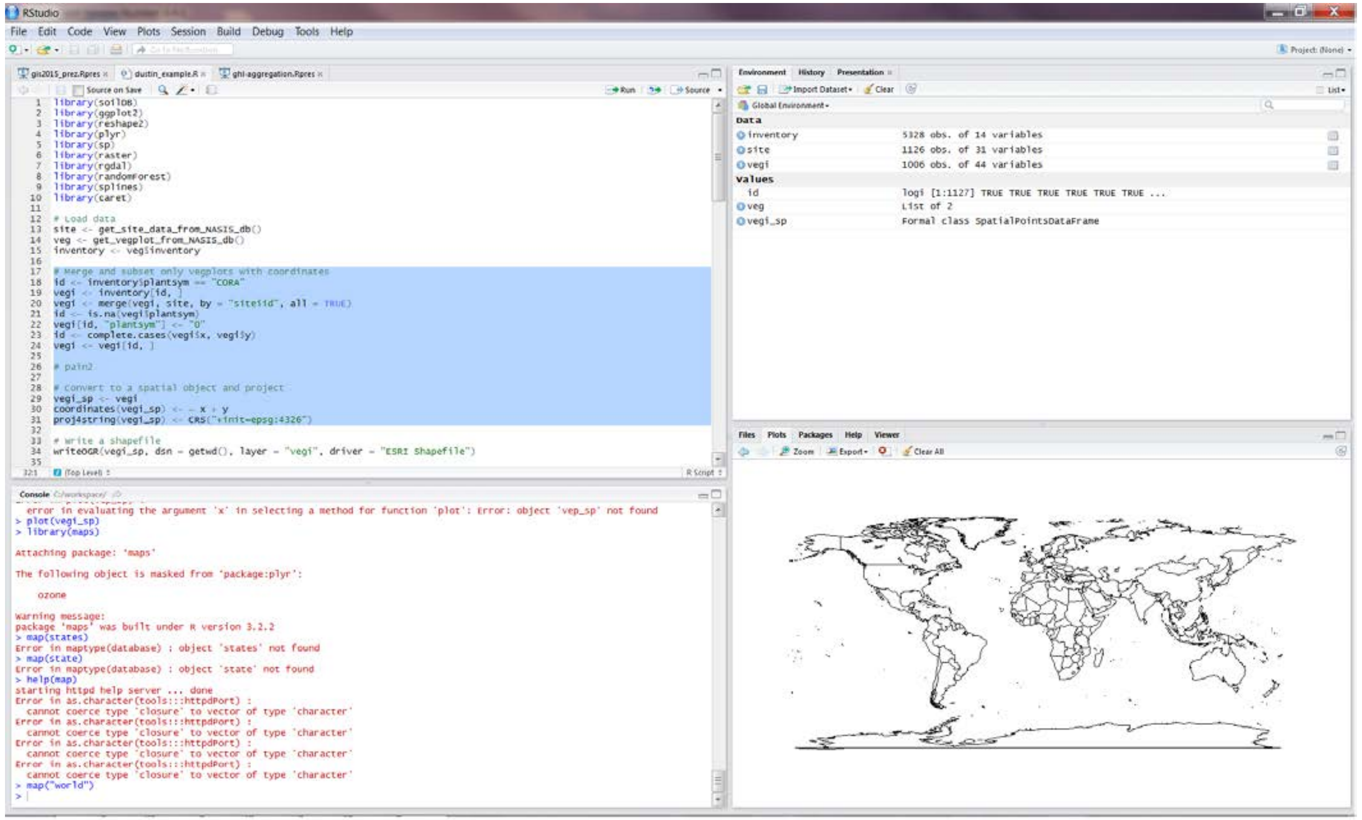
Stephen Roecker
Soil Scientist & GIS Specialist
USDA-NRCS Soil Science Division
Indianapolis

WHAT IS R?

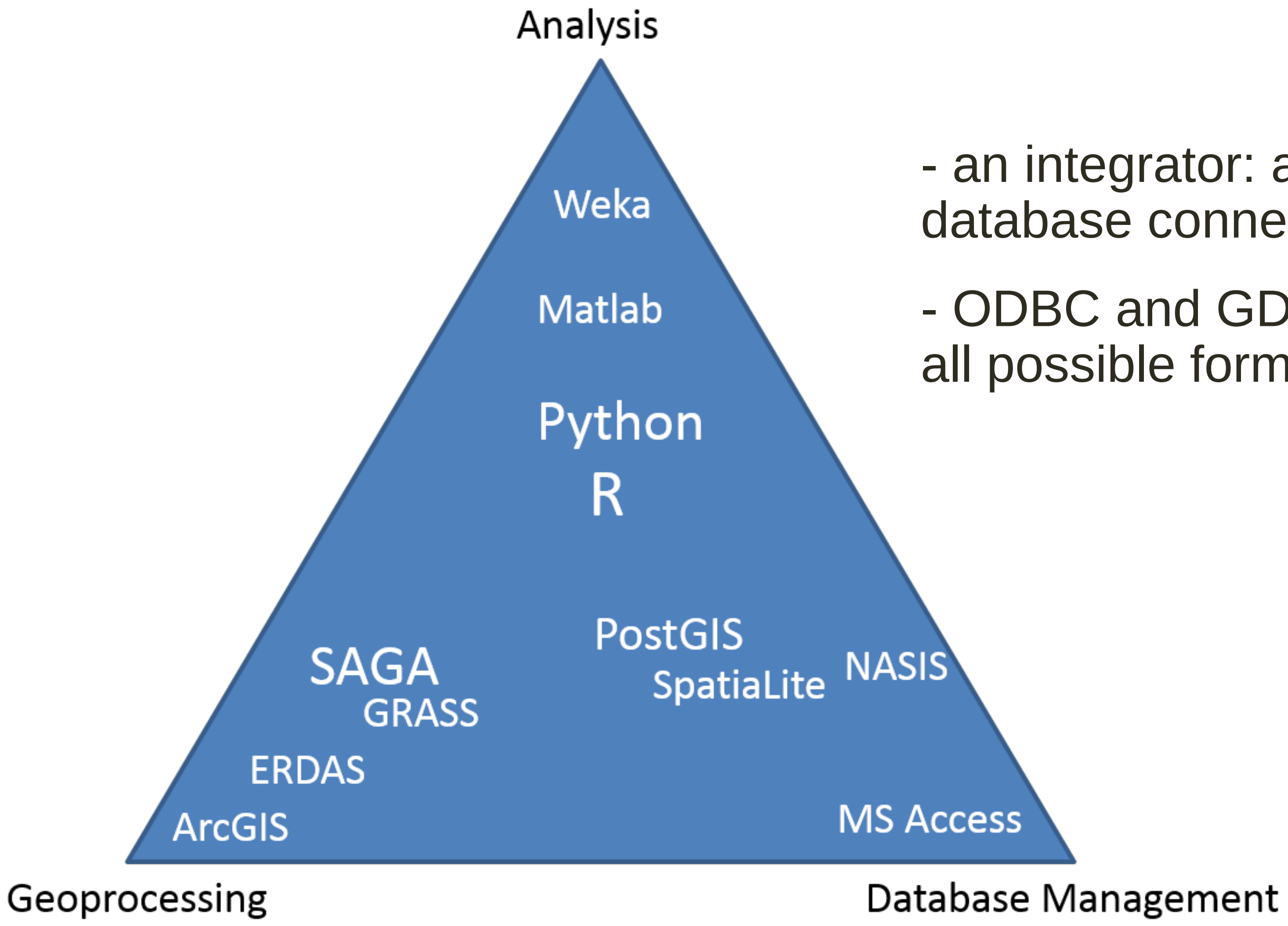
- an analysis platform: calculator, statistics, GIS, Remote Sensing, Raster Modeling, etc...

- programming language: object oriented

- reproducible research: text + code + graphics



WHAT IS R?



- an integrator: analysis + GIS + database connectivity

- ODBC and GDAL link R to nearly all possible formats/interfaces

R PACKAGES

R spatial packages

- rgdal - importing/exporting
- proj4 – projections
- sp - vector and raster processing
- rgeos – vector processing
- raster - raster processing (on disk)

R packages that interface real GIS

- GDAL via rgdal or gdalUtils
- GRASS via spgrass6
- SAGA via RSAGA
- ArcGIS via RPyGeo

ArcGIS extensions for R

- Geospatial Modeling Environment (formerly Hawth's Analysis tools for ArcGIS)
- R-ArcGIS on Github https://r-arcgis.github.io/

BTW - Microsoft just bought Revolution Analytics (an R company)

R BASICS

```r
y <- 2 + 2
y # or print(y)
```

```
[1] 4
```

```r
y <- c(1:10)
mean(y)
```

```
[1] 5.5
```

```r
help(mean) # ?mean
```

R BASICS

mean {base}

R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

`x`
: An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.

`trim`
: the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

`na.rm`
: a logical value indicating whether `NA` values should be stripped before the computation proceeds.

POINT DATA

```
library(sp)
data(meuse) # dataset from Burrough and
McDonnell (1998)
# readOGR() read vector data
# readGDAL() or raster() read raster data
meuse[1:5, 1:6]
```

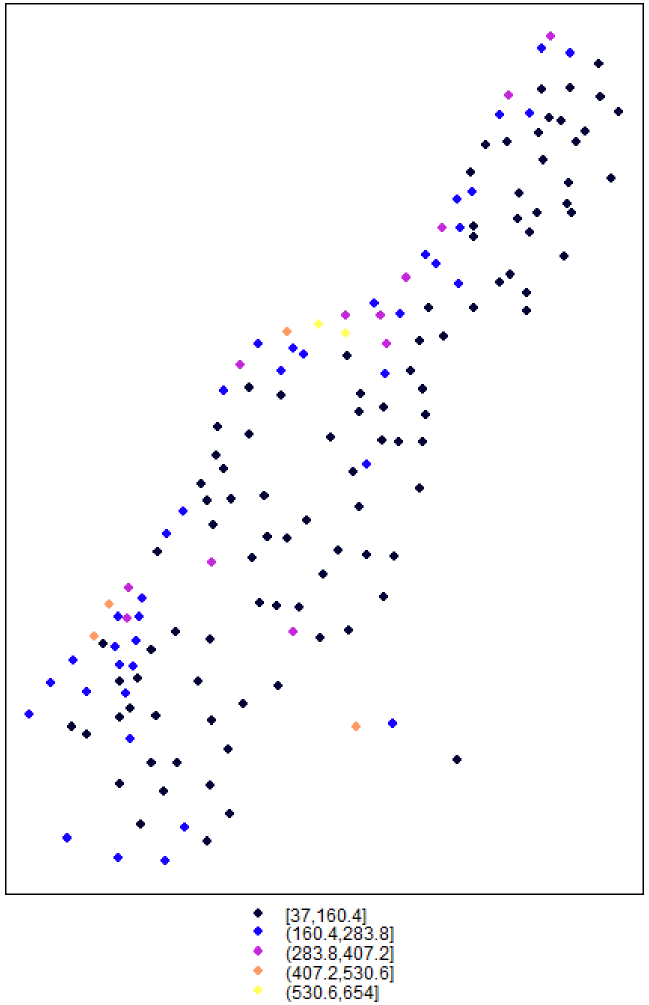
```
        x      y cadmium copper lead zinc
1 181072 333611    11.7     85  299 1022
2 181025 333558     8.6     81  277 1141
3 181165 333537     6.5     68  199  640
4 181298 333484     2.6     81  116  257
5 181307 333330     2.8     48  117  269
```

POINT DATA

```
meuse_sp <- meuse
coordinates(meuse_sp) <- ~ x + y
proj4string(meuse_sp) <-
CRS("+init=epsg:28992")
str(meuse_sp, max.level = 2)
```

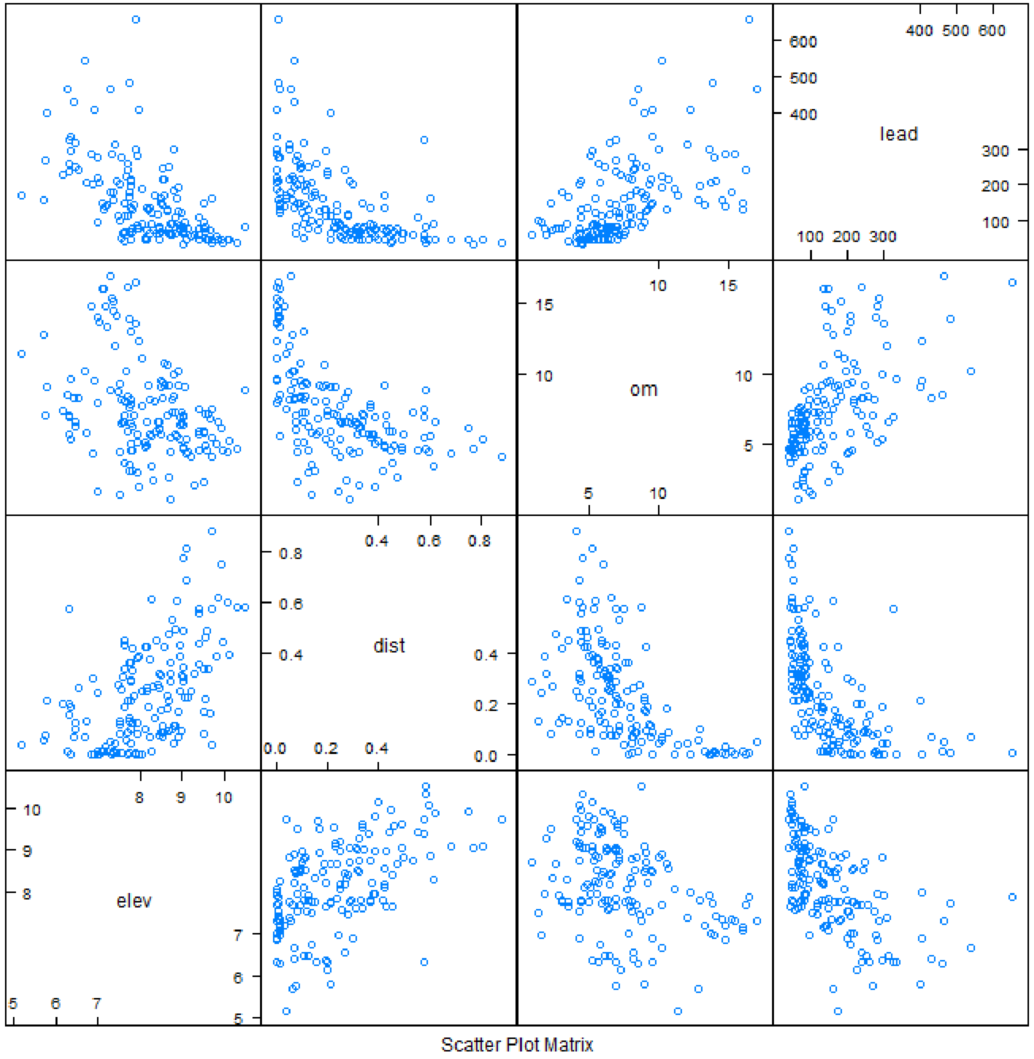
```
Formal class 'SpatialPointsDataFrame'
[package "sp"] with 5 slots
  ..@ data       :'data.frame': 155 obs. of
12 variables:
  ..@ coords.nrs : int [1:2] 1 2
  ..@ coords     : num [1:155, 1:2] 181072
181025 181165 181298 181307 ...
  .. ..- attr(*, "dimnames")=List of 2
  ..@ bbox       : num [1:2, 1:2] 178605
329714 181390 333611
  .. ..- attr(*, "dimnames")=List of 2
  ..@ proj4string:Formal class 'CRS'
[package "sp"] with 1 slot
```

```
spplot(meuse_sp, zcol = "lead")
```



POINT DATA

```
library(lattice)
var <- c("elev", "dist", "om", "lead")
splom(meuse[var])
```



YOU LOST ME!

"Why would I want to code my GIS work, that's why I never bothered to learned GRASS?“

1. Automate/Reduce Repetition (oh yeah)
    1. Downloading data
    2. Geoprocessing data
    3. Analyzing data
2. Reproducible Research (the foundation of SCIENCE!!!)
    1. Documenting your steps (its more compact)
    2. Sharing your work https://github.com/ncss-tech

SAGA AND R

```
library(RSAGA)
rsaga.get.libraries()[63:68]
```

```
[1] "ta_hydrology"          "ta_lighting"
"ta_morphometry"
[4] "ta_preprocessor"       "ta_profiles"
"ta_slope_stability"
```

```
rsaga.get.modules("ta_morphometry")[[1]][c(1,
8, 24, 26), 2]
```

```
[1] Slope, Aspect, Curvature
[2] Morphometric Protection Index
[3] Morphometric Features
[4] Fuzzy Landform Element Classification
27 Levels: Convergence Index ... Wind Effect
(Windward / Leeward Index)
```

SAGA AND R

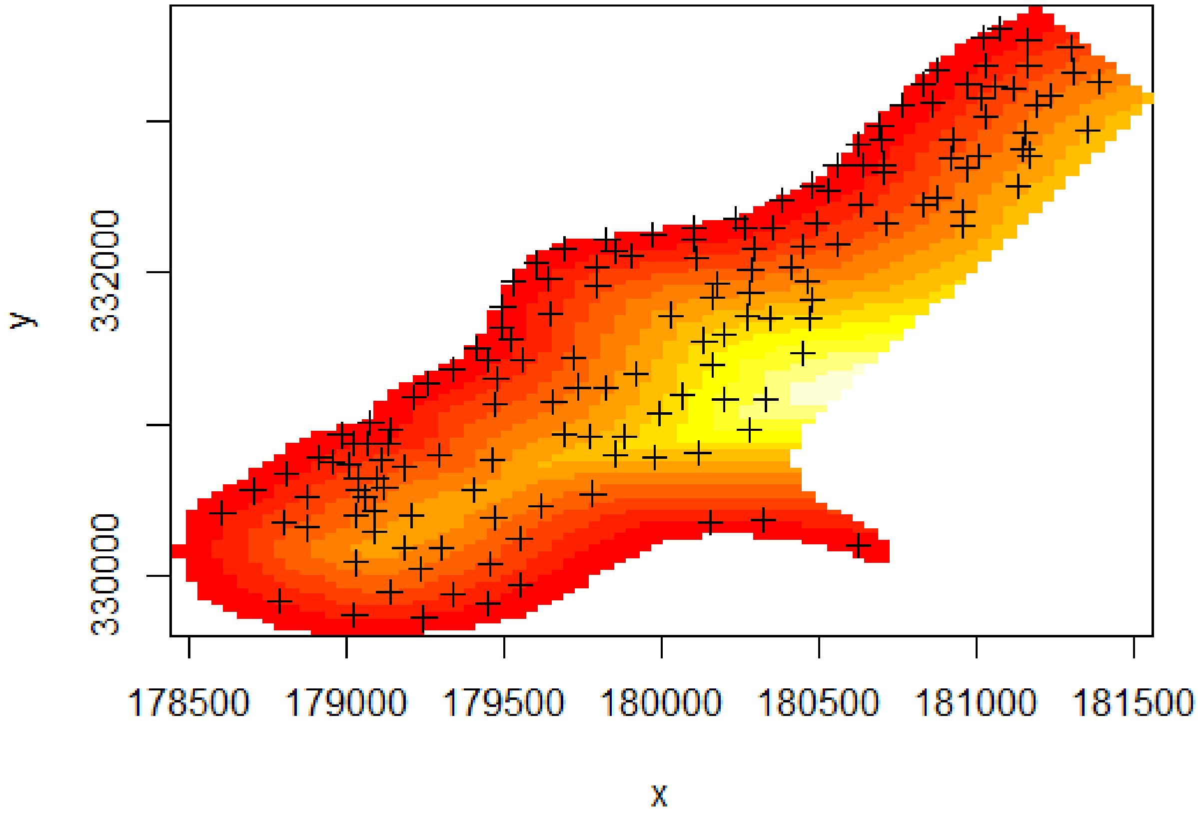
```
rsaga.get.usage("ta_morphometry", 1)
```

```
library path:
C:\Users\Stephen\DOCUME~1\R\WIN-
LI~1\3.2\RSAGA\SAGA-GIS\modules\
library name:    ta_morphometry
library     :    Morphometry
Usage: saga_cmd ta_morphometry 1 -ELEVATION
<str> [-RESULT <str>] [-METHOD <str>] [-
NEIGHBOURS <str>]
  -ELEVATION:<str>  Elevation
    Grid (input)
  -RESULT:<str>     Convergence Index
    Grid (output)
  -METHOD:<str>     Method
    Choice
    Available Choices:
    [0] Aspect
```

RASTER MODELING IN R

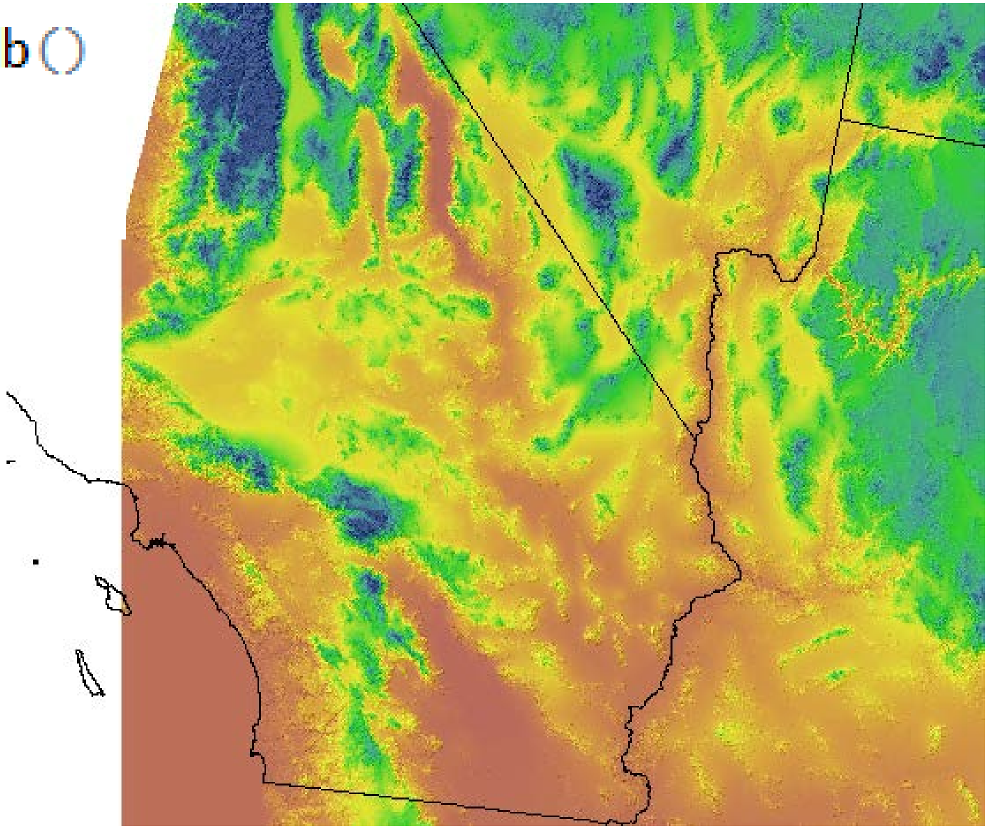
Summary

1. Collect point data
2. Prep point data
3. Extract intersection of point data with other ancillary spatial data
4. Prep point and ancillary data
5. Explore the data
6. Fit a statistical model
7. Predict the model spatially



RASTER MODELING IN R

```
# Load data
site <- get_site_data_from_NASIS_db()
veg <- get_vegplot_from_NASIS_db()
inventory <- veg$inventory
```

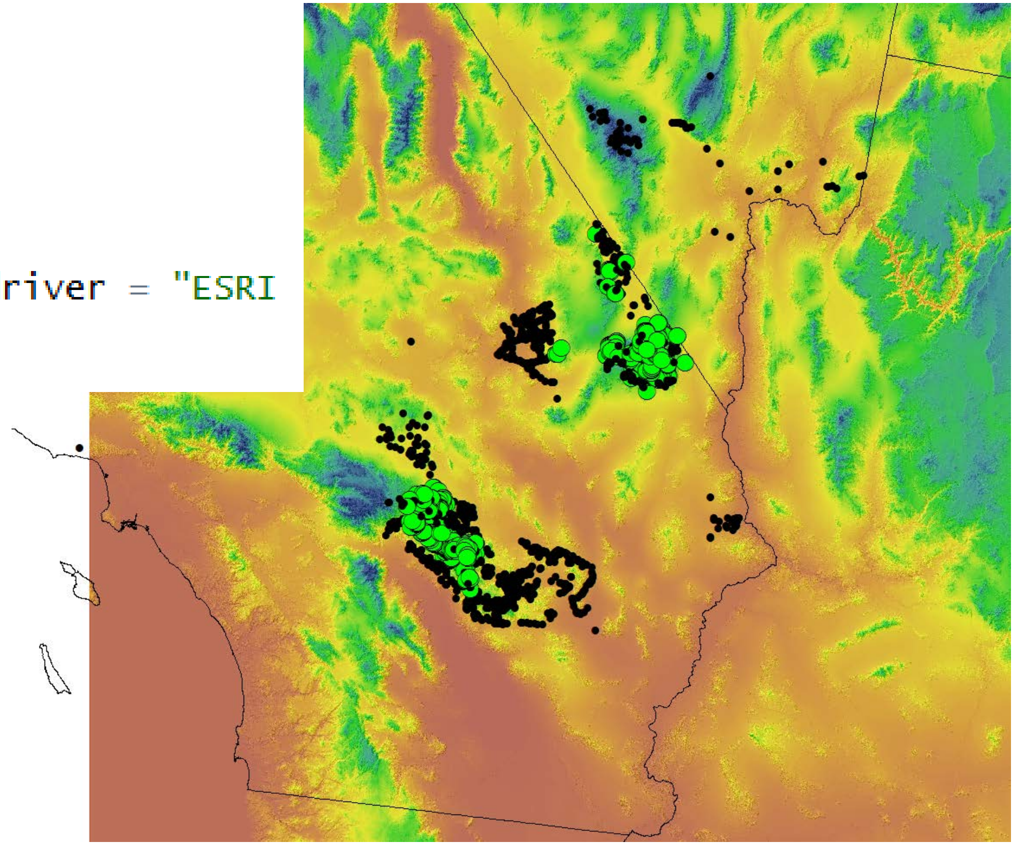


RASTER MODELING IN R

```r
# Merge and subset only vegplots with coordinates
id <- inventory$plantsym == "YUBR"
vegi <- inventory[id, ]
vegi <- merge(vegi, site, by = "siteiid", all = TRUE)
id <- is.na(vegi$plantsym)
vegi[id, "plantsym"] <- "0"
id <- complete.cases(vegi$x, vegi$y)
vegi <- vegi[id, ]
```

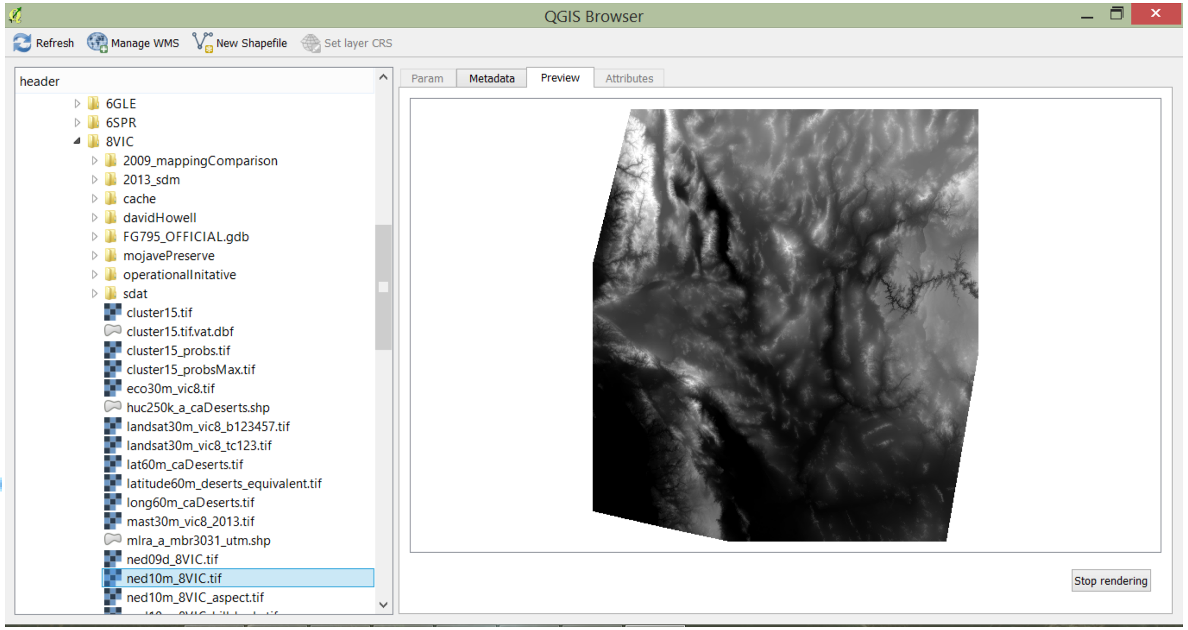
RASTER MODELING IN R

```
# Convert to a spatial object and project
vegi_sp <- vegi
coordinates(vegi_sp) <- ~ x + y
proj4string(vegi_sp) <- CRS("+init=epsg:4326")

# Write a shapefile
writeOGR(vegi_sp, dsn = getwd(), layer = "vegi", driver = "ESRI
Shapefile")
```



RASTER MODELING IN R

```r
# Load geodata
setwd("M:/geodata/project_data/8VIC")

grid.list <- c(
  "mast30m_vic8_2013.tif",
  "prism30m_vic8_tavg_1981_2010_annual_C.tif",
  "prism30m_vic8_ppt_1981_2010_annual_mm.tif",
  "prism30m_vic8_ppt_1981_2010_summer_mm.tif"
)

geodata <- stack(grid.list)
names(geodata) <- c("mast", "maat", "map", "msp")

# Join veg and geodata
geo <- extract(geodata, vegi_sp, df=T)
vegig <- cbind(data.frame(vegi), geo)
vegig <- vegig[, c("plantsym", "mast", "maat", "map", "msp")]
vegig <- na.exclude(vegig)
vegig$plantsym <- as.factor(vegig$plantsym)
```

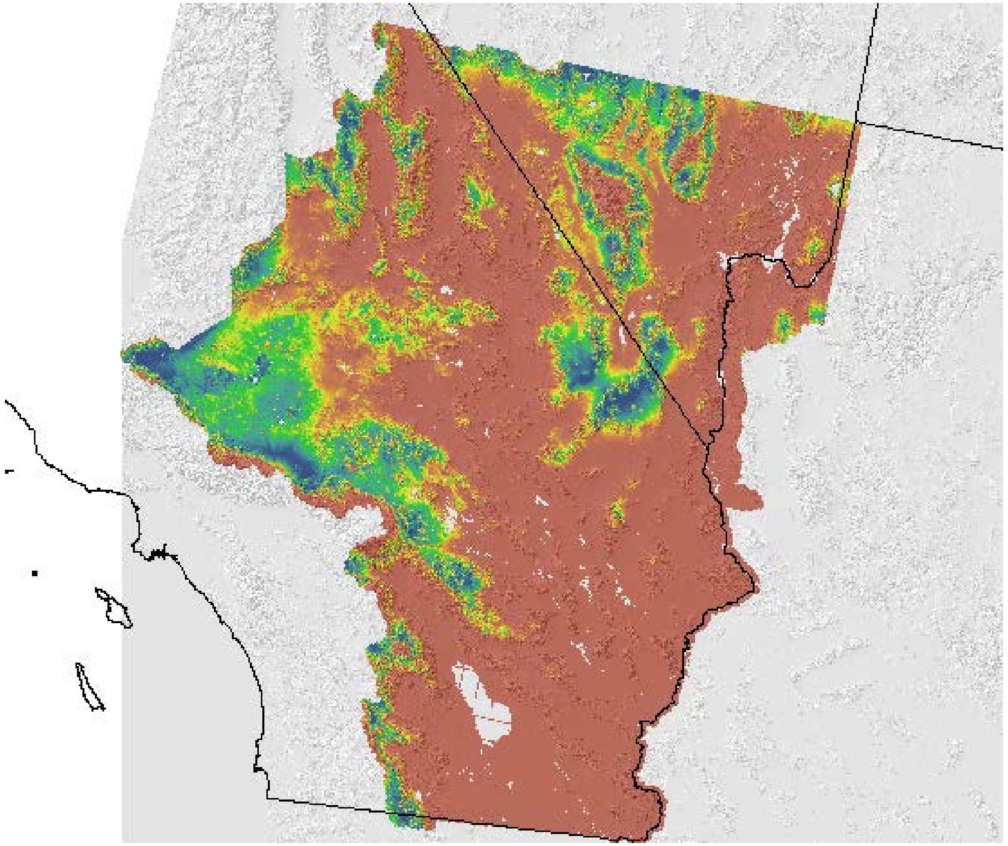


RASTER MODELING IN R

```
# Fit vegi GLM
vegi_glm <- glm(plantsym ~ maat + ns(map, 2) + msp, data = vegig,
family = binomial)

summary(vegi_glm)

confusionMatrix(vegi_glm$y > 0.5, predict(vegi_glm, type =
"response") > 0.5)
```

```
# Apply vegi GLM to raster stack
predfun <- function(model, data) {
  v <- predict(model, data, type="prob")
}

vegi_raster <- predict(geodata, vegi_rf, fun = predfun, index = 1,
progress = "text")

writeRaster(vegi_raster, filename = "M:/geodata/vegi_raster.tif"
,format = "GTiff", datatype="FLT4S", overwrite=T, NAflag = -99999,
progress = "text")
```

RASTER MODELING IN R



REPORTING CAPABILITIES

https://github.com/ncss-tech/soil-pit/tree/master/examples

R GIS RESOURCES

Websites

- CRAN Spatial View - https://cran.r-project.org/web/views/Spatial.html

Mailing list

- R-SIG-Geo (be sure and ask nice)
- stackoverflow

Books

- A Practical Guide to Geostatistical Mapping
- Applied Spatial Data Analysis with R
- Learning R for Geospatial Analysis
- An Introduction to R for Spatial Analysis & Mapping

QUESTIONS

?