CLASS NOTES FOR
INTRODUCTION TO GEOSTATISTICS: THEORY & APPLICATIONS IN R

A two-day short course presented to the Dept of Civil & Env Engineering at the University of Strathclyde, 7 – 8 November 2019

by Ryan M. Pollyea
Virginia Tech Dept. of Geosciences

Exploratory Data Analysis

Open juradata_r.xlsx and save comma delimited file (juradata_r.csv)

For description of the Jura Data set, see Ch. 1.1 in Geostatistics for Natural Resource Evaluation by P. Goovaerts, Oxford University Press, 1997.

Start R session & navigate to directory with juradata_r.csv

```
> setwd("/path/to/directory/with/juradata_r.csv")
```

Load the Jura data set.

```
> jura.data <- read.csv("juradata_r.csv", header=T)
```

Load sp & gstat library:

```
> library(sp)
> library(gstat)
```

Convert to spatial data frame by assigning coordinates:

```
> coordinates(jura.data) = ~X+Y
```

See that jura.data is a spatial object:

```
> class(jura.data)
```

Begin Exploratory Data Analysis

```
> summary(jura.data)
> mean(jura.data$Cd) note that this is n-1
> sd(jura.data$Cd) note that this is n-1
```

To see frequency distribution – this is the total in each bin, bars sum to N

```
> hist(jura.data$Cd)
```

Show that bars sum to N:

```
> jura.cd.hist <- hist(jura.data$Cd, plot=F)
> sum(jura.cd.hist$counts)
```

To see probability density function – this is the integral form, area of bars sums to 1.

```
> hist(jura.data$Cd, freq=F)
```

To see relative frequency distribution – sum of bars is 1 – for large N, approximates prob distribution:

```
> library(HistogramTools)
> PlotRelativeFrequency(hist(jura.data$Cd))
```

Begin looking at spatial distribution of data.
Bubble plot:

```
> bubble(jura.data, "Cd")
```

Bubble plot w/ axes

```
> plot(jura.data, pch=1, cex=jura.data$Cd/2, axes=T)
> cd.leg = c(1, 2, 3, 4)
> legend("topleft", legend=cd.leg, pch=1, pt.cex=cd.leg/2)
```

Raw Semivariogram

```
> cd.var.raw = variogram(Cd~1, jura.data, cloud=TRUE)
> plot(cd.var.raw)
```

Sample/Experimental/Empirical Variogram

```
> cd.var = variogram(Cd~1, jura.data, width=0.15)
> plot(cd.var)
```

→ Use "width" in distance units to refine or coarsen variogram discretization
https://www.rdocumentation.org/packages/gstat/versions/2.0-0/topics/variogram

Model Variogram

```
> cd.vmod = vgm(psill=0.5*var(jura.data$Cd), "Exp", range=0.3,
nugget=0.5*var(jura.data$Cd))
```

Note: vgm(psill = X, "Sph, Exp, Gau", range=X, nugget=X, anis=c(p,q,r,s,t))
https://www.rdocumentation.org/packages/gstat/versions/2.0-0/topics/vgm

Check Sill of model variogram:

```
> sum(cd.vmod$psill)
> var(jura.data$Cd)
```

Plot experimental & model variograms

```
> plot(cd.var, cd.vmod)
```

Kriging w/ GSTAT

1. Make grid & create polygonal border for site from set points.

Open jura_border.xlsx and save as jura_border.csv

```
> jura.border = read.csv("jura_border.csv", header=T)
> coordinates(jura.border) = ~X+Y
> tmp1 = Polygon(jura.border)
> tmp2 = Polygons(list(tmp1),1)
> jura.b = SpatialPolygons(list(tmp2))
> plot(jura.b)
```

Ordinary kriging (default)

Krige onto Jura grid.

```
> jura.grid = spsample(jura.b, n=1000, "regular")
> gridded(jura.grid) = TRUE

> cd.krg = krige(Co~1, jura.data, jura.grid, model=cd.vmod)
> spplot(cd.krg["var1.pred"], scales=list(draw=TRUE))
```

Cross-Validation

```
> cd.xval <- krige.cv(Cd~1, jura.data, jura.grid, model=cd.vmod)
```

Check minimum variance

```
> mean(cd.xval$residual)
```

Check normally distributed residuals

```
> hist(cd.xval$residual)
```

Change co.xval to spatial points data frame

```
> coordinates(cd.xval) = ~X+Y
```

Check whether residuals are correlated w/ sample vargioram

```
> plot(variogram(residual~1, data=cd.xval))
```

Anisotropy – Use Cobalt for anisotropy

Variogram Map

```
> co.vmap = variogram(Co~1, jura.data, map=TRUE, cutoff=2,
width=0.125)
cutoff = size of ½ map; width = size of pixel
> plot(co.vmap, main="Cobalt Variogram Map")
```

Anisotropic Variogram

Experimental

```
> co.var70 <-variogram(Co~1, jura.data, alpha=70, tol.hor=11.25,
width=0.2, cutoff=2.5)
> co.var160 <-variogram(Co~1, jura.data, alpha=160,
tol.hor=11.25, width=0.2)
```

To see both empirical variograms on same plot.

```
> plot(co.var160$dist, co.var160$gamma, col="red")
> points(co.var70$dist, co.var70$gamma, col="blue")
```

Model

```
> co.vmod70 <- vgm(psill=0.75*var(jura.data$Co), "Sph",
range=2.25, nugget=0.25*var(jura.data$Co))
> co.vmod160 <- vgm(psill=0.75*var(jura.data$Co), "Sph", range=1,
nugget=0.25*var(jura.data$Co))
```

Put complete anisotropic model into a single object:

```
co.anisomod <- vgm(psill=0.75*var(jura.data$Co), "Sph",
range=2.25, nugget=0.25*var(jura.data$Co), anis=c(70,0.45))
```

Kriging with Anisotropy

Ordinary kriging

```
> co.krg.anis = krige(Co~1, jura.data, jura.grid,
model=co.anisomod )
> spplot(co.krg.anis["var1.pred"], scales = list(draw = TRUE))
```

Conditional Simulation

```
> co.csim = krige(Co~1, jura.data, jura.grid, model=co.vmod,
nmax=30, nsim=4)
> spplot(co.csim, main="Four Conditional Simulations")
```

Sequential simulation (conditional) w/ Anisotropy

```
> co.krg.csim = krige(Co~1, jura.data, jura.grid,
model=co.anisomod, nsim=4, maxdist=1)
> spplot(co.krg.csim, scales = list(draw = TRUE))
```

E-Type Estimates (mean & standard deviation)

→ also, this is not particularly elegant, but it works.

Create a dataframe with just simulation results

```
> co.simdat <- data.frame(co.krg.csim$sim1, co.krg.csim$sim2,
co.krg.csim$sim3, co.krg.csim$sim4)
```

Calculate row means for each pixel:

```
> co.means <- rowMeans(co.simdat)
```

Calculate row std dev for each pixel:

```
> library(Rfast)
> co.stdev <- rowVars(as.matrix(co.simdat), std=T)
> co.stdev <- data.frame(co.stdev)
```

Make data frame with X, Y, row means & row std dev

```
> co.etype <- data.frame(co.krg.csim$x1, co.krg.csim$x2,
co.means, co.stdev)
> coordinates(co.etype) = ~co.krg.csim.x1+co.krg.csim.x2
> gridded(co.etype) = TRUE
> spplot(co.etype, scales = list(draw = TRUE))
```

→ Note that I'm using standard deviation for uncertainty b/c same units as data, therefore can plot on the same scale bar!