Atmospheric Data - STAT 265 Final (Enter your own title)

Background

Data from NASA Langley Research Center Atmospheric Sciences

http://www.srh.noaa.gov/srh/jetstream/clouds/cloudwise/types.html

We have atmospheric measures on a very coarse 24 by 24 grid covering Central and a part of South America. The longitude ranges from 113.8 W to 56.2 W and latitude from 36.2 N to 21.2 S. There are several variables that each of you will examine separately: *Ozone* is measured in Dobson Unit (defined to be 0.01 mm thickness at 0 degree celcius and standard air pressure of 1 bar); *Sea surface temperature* (SST) *air temperature* are measured in celcius; *Cloud cover* for low, mid, and high levels is measured in percentage where mid level ranges from 2 - 8 kilometers (6500 - 16500 feet) above ground; *Pressure* is measured in millibars. The data for all variables contain monthly average for Dec 2000.

Objectives

Learn the options and the steps of functional applications in R.

1. Run a bi-linear interpolation (linear in both x and y directions) to see what the data look like.
2. Learn about two additional 3D plotting tools in R: a dropline plot and a drape plot (color-added perspective plot). And play with the color spectrum.
3. Perform kriging and explain what the output suggests.

Expectation

Provide a 1-3 page written report. Include no more than four plots. Each figure should be less than 4 inches tall and be appropriately titled. Adding a caption is optional. Submit the report via e-mail with the Rmd file.

- Describe the pattern in the data using the straight-up interpolation and the kriging output.
- What are the obvious patterns that you wish to incorporate in the final predicted surface?
- Explain the choices you made to fit the variogram: Did you assume a nugget effect? What is the sill and the range of the fitted variogram? Is the *exponential* covariance fucntion appropriate?
- Describe the difference in the usages between the straight-up interpolation and the kriging.

For a **bonus**, fit one of the following covariance fucntions: *Gaussian*, *linear*, or *spherical*, and suggest which one works the best for the continuous process you have.

Submit the project via email by the end of Monday, December 21st. I will respond with the confirmation of receipt.

Please hide the code when knitting the file.

```
######################
## Prepare the Data

## Read in the data set
dat=read.table("cloudmid72-final.txt", header = TRUE, sep = "")
dat <- dat[,-1]
dat <- as.matrix(dat)
hist(dat)

## String the values for the ease of data manipulation
temp <- NULL
  for(i in 1:24) temp <- c(temp, dat[i,])
temp <- unlist(temp)

## Read in the coordinate values from 'ozone1.dat'
dat2=read.table("ozone1.dat")


## Create an array with (x,y,z) columns
dat <- cbind(dat2[,1:2], temp)

## Add column names to 'dat'
# colnames(cloud) <- c("long", "lat", ??)

## Save the data and check it works.
## The file name should be different from the data file.
# write.table(dat, file = "??.txt", sep = " ")
# dat <- read.table("??.txt", as.is=TRUE)



########################
## 1. Bi-linear interpolation

## Load library for interpolation
library(akima)

dat2.interp = interp(dat2[,1],dat2[,2],dat2[,3])
plot.surface(dat2.interp)
## The 'interp' function works for irregularly-space data.


attach(dat)
# dat.interp = interp(long, lat, ??)
plot.surface(dat.interp)
```

```
########################
## 2. 3D plots

## Identify the color palette that best fits the data you represent.
## First load the appropriate library.
# library(??)
```

```
## Then, change the color spectrum in 'brewer.pal' below.


## Dropline plot
library(scatterplot3d)
scatterplot3d(dat, type='h', cex.lab=1.3,
              xlab="longitude",ylab="latitude",zlab="dat level")

## Perspective-drape plot
library(fields)
plot.surface(dat.interp, type="p")

## An image with a legend
plot.surface(dat.interp, type="I", col=brewer.pal(9, "PuOr")) # change color

## Overlay the world map
library(maps)
map(add=T, lwd = 2)

## Contours on image
plot.surface(dat.interp, type="C", col=brewer.pal(9, "PuOr")) # adjust color

# Show two plots at once: drape & image
plot.surface(dat.interp, type="b", col=brewer.pal(9, "PuOr")) # adjust color
```

```
#########################
## 3.1 Variogram for a continuous process data

library(geoR)
## Assume a constant mean: mu(x,y) = mu

# variog0.dat=variog(coords = dat[,c(1,2)], data = ??,
                estimator.type="modulus", trend="cte", max.dist=40)
plot(variog0.dat)

## Assume a linear trend (1st) in the mean structure
## mu(x,y) = beta0 + beta1*x + beta2*y

# variog1.dat=variog(coords = dat[,c(1,2)], data = dat[??],
                estimator.type="modulus", trend="1st", max.dist=40)
plot(variog1.dat)

## Assume a quadratic trend (2nd) in the mean structure
## mu(x,y)= beta0 + beta1*x + beta2*y + beta3*x^2 + beta4*y^2 + beta5*x*y.

# variog2.dat=variog(coords = dat[,c(1,2)], data = dat[??],
                estimator.type="modulus", trend="2nd", max.dist=40)
plot(variog2.dat)



## Obtain weighted least squares estimate of parameters
## for an exponential variogram assuming an appropriate trend level
```

```
# vario.fitted = variofit(??, fix.nug=TRUE, cov.model="exponential")

## What are thr parameter estimates?
vario.fitted




### Examine anisotropy:
plot(variog4(coords=dat[,c(1,2)], data=dat[,3], trend="1st", max.dist=40))
plot(variog4(coords=dat[,c(1,2)], data=dat[,3], trend="2nd", max.dist=40))

## If you are convinced to include 'anisotropy' to describe
## the process (which isn't true for all cases), then run the following line.
## NOTE: it will take at least 20-30 seconds to run
## and it will spit out some automatic messages and some
## warning. But run the fourth line below to make sure
## the parameters are estimated.

dat.reml2 = likfit(coords = dat[,c(1,2)], data = dat[??],
                fix.nugget=T, ini=c(.5,.5), cov.model="exponential",
                trend ="1st", method='REML',
                fix.psiA = FALSE, fix.psiR = FALSE)
dat.reml2
```

We want to make a smoothed prediction map using kriging predictions over a dense grid. Then we will draw pictures based on these predictions.

```
##############################################################################
## 3.2 Kriging
## After fitting the variogram, we have estimates of covariance parameters.
## We can now predict at new locations based on the fitted variogram values.



## Creating a 50 x 50 grid using 'xvec' and 'yvec' below
xvec = seq(min(dat[,1]), max(dat[,1]), len = 50)
yvec = seq(min(dat[,2]), max(dat[,2]), len = 50)
grid50 = expand.grid(xvec, yvec)

## Kriging prediction
## Note 1. Fill in the "cov.pars" argument as well as the "cov.model".
## Note 2. The trend is set at '1' assuming you chose a linear trend.
## If you chose a quadratic trend, set it at '2'.
## If you chose a constant trend, then completly remove m0 = "kt" field.
## Note 3. If you decided to describe the process with anisotropy
## then supply two parameter values 'psiA (angle)' and 'psiR (ratio)'
## as concatenated arguments to 'aniso.pars'.

# dat.grid50 =ksline(coords = dat[,c(1,2)], data = dat[,3],
                  # locations = grid50, cov.pars=c(??, ??),
                  # cov.model = "??", m0="kt", trend = 1,
                  ktedata = NULL, ktelocations = NULL,
```

```
                  aniso.pars = NULL, signal = FALSE, dist.epsilon = 1e-10, messages)

dat.krige = interp(grid50[,1], grid50[,2], dat.grid50$predict)

# plot.surface(dat.krige, type="C", main="INCLUDE TITLE HERE")
map(add=T, col = "pink")

dat.krigeSD=interp(grid50[,1], grid50[,2], sqrt(dat.grid50$krige.var))
# plot.surface(dat.krigeSD, type="C", main="INCLUDE TITLE HERE")


####### THE END ########
```

Have a wonderful winter break!