

Parametric/Median Polish Trend Removal

Before studying models for variograms (which describe small-scale or residual variation), we discuss the removal of any large-scale variation (or trend) that might be present. In spatial modeling, researchers have historically taken two different views on describing any trend present. Both of these views recognize the same basic model form for the process. Before discussing the two approaches taken toward modeling any trend present, it is important to understand the following model form used in virtually any type of nondeterministic model.

Model Form: Recall that the set of random variables $\{Y(\mathbf{s}) : \mathbf{s} \in \mathcal{R}\}$ is a random function, with some mean and variance given by $\mu(\mathbf{s})$ & $\sigma^2(\mathbf{s})$ respectively. Under weak stationarity, the mean and variance are no longer functions of the location \mathbf{s} . However, more generally (and realistically), nondeterministic models are assumed to have the form:

$$\begin{aligned} Y(\mathbf{s}) &= \mu(\mathbf{s}) + \epsilon(\mathbf{s}), & \mathbf{s} \in \mathcal{R}, \text{ the region of interest.} \\ &= \begin{pmatrix} \text{Large-scale} \\ \text{variation} \end{pmatrix} + \begin{pmatrix} \text{Small-scale} \\ \text{variation} \end{pmatrix} \end{aligned}$$

- The mean (or trend) function $\mu(\mathbf{s})$ may be a function of the location and/or of some list of covariates. It may be expressed as a linear or nonlinear function of some regression parameters, a generalized linear model (GLM), a generalized additive model (GAM), or most generally may be modeled nonparametrically. One of the goals of this handout is to introduce a nonparametric method of modeling the trend known as median polish.

As mentioned at the outset, there are two general approaches taken toward modeling the trend and error structure in spatial processes. These are:

1. First remove the trend and perform any subsequent analysis on the resulting residuals.
2. Model the trend and error structure simultaneously. (One example of this approach is universal kriging, discussed later in the course.)

We will take the first approach for now, and look at both parametric and nonparametric methods of trend removal.

Example: Consider the following data on arsenic concentrations taken over 21 locations with corresponding elevations. Suppose we let:

$$\begin{aligned} z &= \text{the arsenic concentration measurements (in ppb),} \\ w &= \text{the corresponding elevations (in feet),} \\ x &= s_1 = \text{the x-coordinate of the measurement, and} \\ y &= s_2 = \text{the y-coordinate of the measurement.} \end{aligned}$$

1. If there were a roughly linear trend in the x-direction, and a linear relationship between arsenic levels and elevations, we might try modeling the mean function with the regression model:

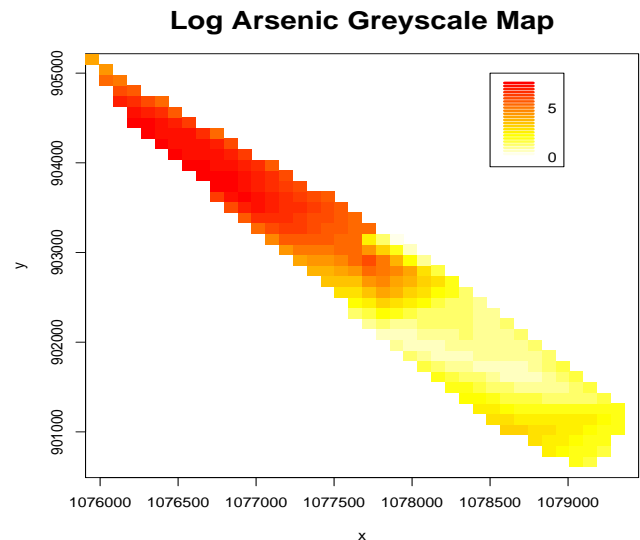
2. Suppose there were linear trends in both the x- and y-directions which were interactive, and an exponentially decreasing relationship between arsenic levels and elevation. What mean model might we try fitting to these data?

3. A greyscale plot (shown to the right) revealed a definite SE-NW trend in the arsenic data. How might we remove this trend with a regression model on the mean function?

Assuming that elevation is not related to log arsenic concentration level, we might fit the model:

$$\mu(\mathbf{s}) = \beta_0 + \beta_1 x(\mathbf{s}) + \beta_2 y(\mathbf{s}).$$

Fitting this in **R** via the **lm** command as given below, gave the output which follows:



```
arsenic.reg <- lm(log(conc) ~ x+y) # Performs linear regression of the log con-
                                     # centrations on the x- and y-coordinates.
summary(arsenic.reg)               # Summarizes the regression fit.
```

```
Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.710e+03  2.317e+03   1.170    0.257
x            -2.092e-03  1.311e-03  -1.596    0.128
y            -5.004e-04  1.028e-03  -0.487    0.632

Residual standard error: 1.808 on 18 degrees of freedom
Multiple R-squared:  0.4651,    Adjusted R-squared:  0.4057 
F-statistic: 7.826 on 2 and 18 DF,  p-value: 0.003584
```

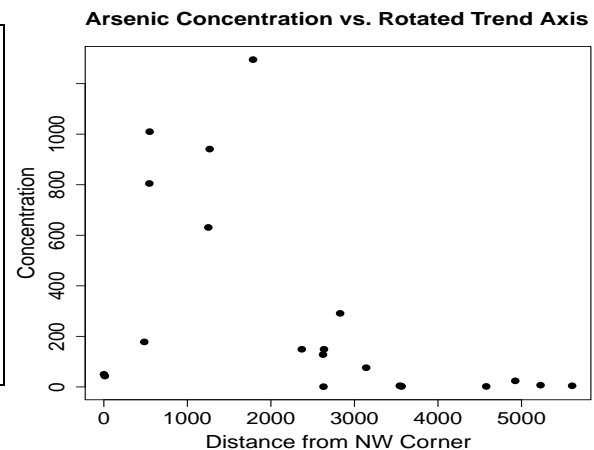
```
cor(x,y)
```

```
[1] -0.957679
```

What does this model fit tell you?

4. Since the trend is really in neither the E-W or N-S direction as x and y are oriented, but in a diagonal direction running from SE-NW, we might be better off creating a new explanatory variable along this direction. For example, since the upper left-hand corner of the region sampled is roughly at coordinates: (1075951, 905157), we might create a variable which is the distance from this point. This was done via **R**, giving a new variable called “newx”. A scatterplot of the arsenic concentrations versus newx was then created to examine the nature of the trend along the NW-SE diagonal. The plot is given to the right, with the corresponding **R** code shown below.

```
newx <- sqrt((x-min(x))**2 +
  (y-max(y))**2) + 1
plot(newx,conc,ylab="Concentration",
  xlab="Distance from NW Corner",
  main="Arsenic Concentration vs.
  Rotated Trend Axis",cex.main=1.6,
  cex.lab=1.6,cex.axis=1.5,cex=1.3,
  mgp=c(2.7,1,0))
```



In viewing this scatterplot, what type of function might you suggest to remove the trend?

Nonlinear least squares was used via the **nls** function in **R** to fit the model given above to these data. The **R** code and output are given below, and the resulting model fit is plotted through the scatterplot for comparison.

```
nlfit <- nls(conc ~ (newx**(alpha-1))*exp(-newx/beta),
  start=list(alpha=2.17, beta=855))
summary(nlfit)
```

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
alpha	2.17681	0.03979	54.709	< 2e-16 ***
beta	718.39128	104.80584	6.854	1.53e-06 ***

Residual standard error: 246.6 on 19 degrees of freedom

Number of iterations to convergence: 9

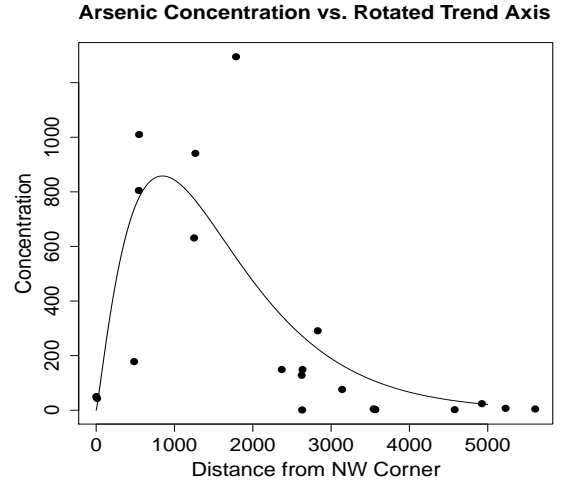
Achieved convergence tolerance: 4.77e-06

```

val <- seq(0,5000,5)
concdens <- (val**
  (summary(nlfit)$coef[1,1]-1))*
  exp(-val/summary(nlfit)$coef[2,1])
lines(val,concdens)

```

- All of what has been considered above in the arsenic example are examples of fitting parametric models to the mean function in an effort to detrend the data *before* attempting any analysis of the spatial correlation structure.



- If there are trends in the data which perhaps are difficult to model in some straightforward functional way, an alternative approach to removing the trend is nonparametric trend removal. One such method which will be discussed in what follows is median polish.

Median Polish: Median polish is the name of an iterative algorithm for removing any trends present by computing medians for various coordinates on the spatial domain \mathcal{R} . This definition will seem vague until the formal algorithm is introduced and run.

- Typically, median polish will be used for lattice data, as will be seen shortly, but one could use these techniques with non-lattice data by overlaying a grid on the observations.

The Algorithm: The median polish algorithm assumes the following decomposition of the random process \mathbf{Z} :

$$Z(\mathbf{s}) = \underbrace{\mu(\mathbf{s})}_{\text{grand effect} + \text{row}(\mathbf{s}) + \text{column}(\mathbf{s})} + \epsilon(\mathbf{s}), \quad (1)$$

and then proceeds by estimating the “grand” effect, the row effects (one for each row), and the column effects (one for each column). The steps of the algorithm are given below:

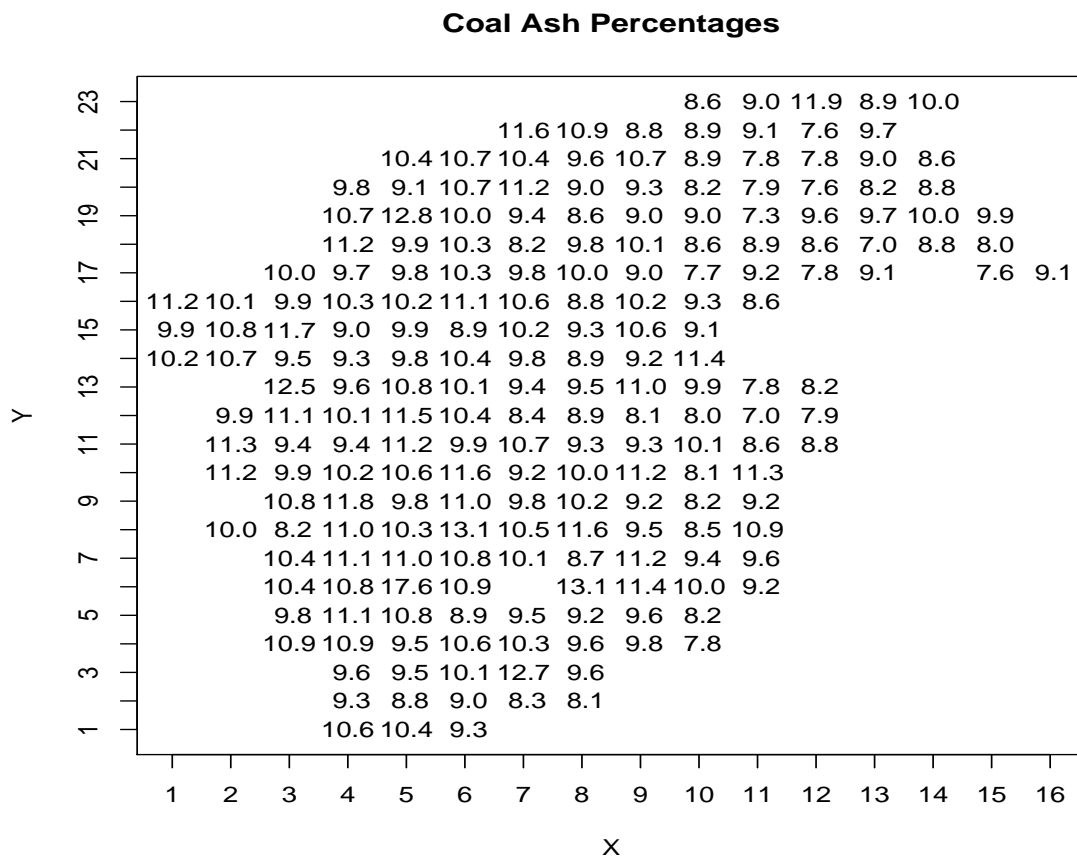
1. Take the median of each row and record the value to the side of the row. Subtract the row median from each point in that particular row.
2. Compute the median of the row medians, and record the value as the grand effect. Subtract this grand effect from each of the row medians.
3. Take the median of each column and record the value beneath the column. Subtract the column median from each point in that particular column.
4. Compute the median of the column medians, and add the value to the current grand effect. Subtract this addition to the grand effect from each of the column medians.
5. Repeat steps 1-4 until no changes occur with the row or column medians.

It should be emphasized that through the course of this algorithm, the relationship given above in (1) between $Z(\mathbf{s})$ and the various median polish effects is preserved at every step.

Example: To see how this works, consider the following 3x3 lattice of magnesium values considered in the discussion of cross-functions. Perform a median polish on these values.

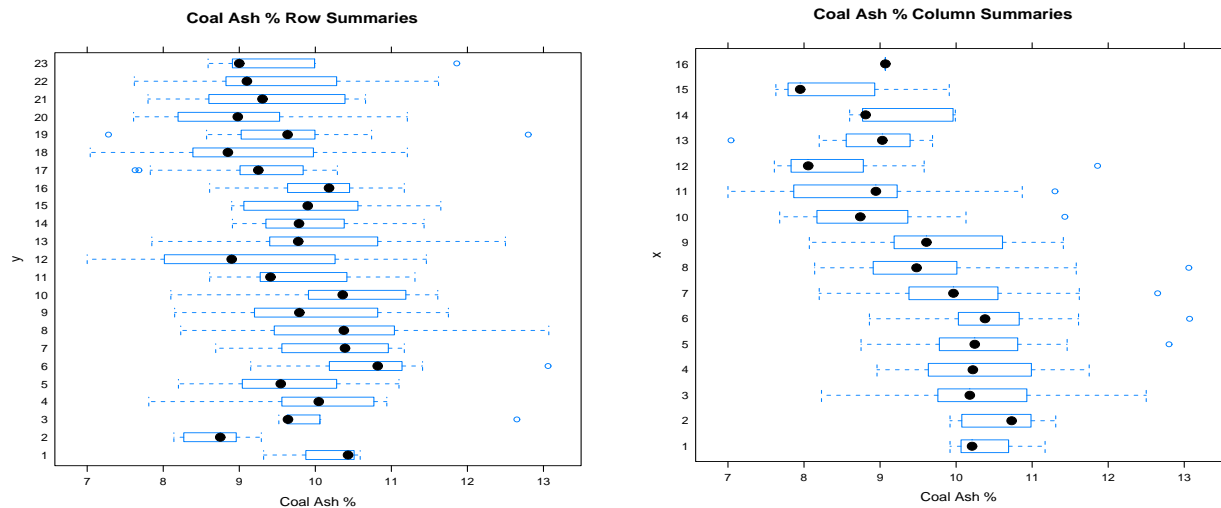
3	4	5
5	4	6
5	6	5

Performing Median Polish in R: To illustrate how to perform median polish in **R**, reconsider the coal ash percentage data from the first week of class. Here, 208 coal ash percentages were measured over the Robena Mine Property in Greene County, PA. A data posting is given below:



- From this initial posting, we noticed that there were some unusually high values in places, and that there seemed to be a trend of larger to smaller coal ash percentages moving roughly from the SW corner to the NE corner of the plot.

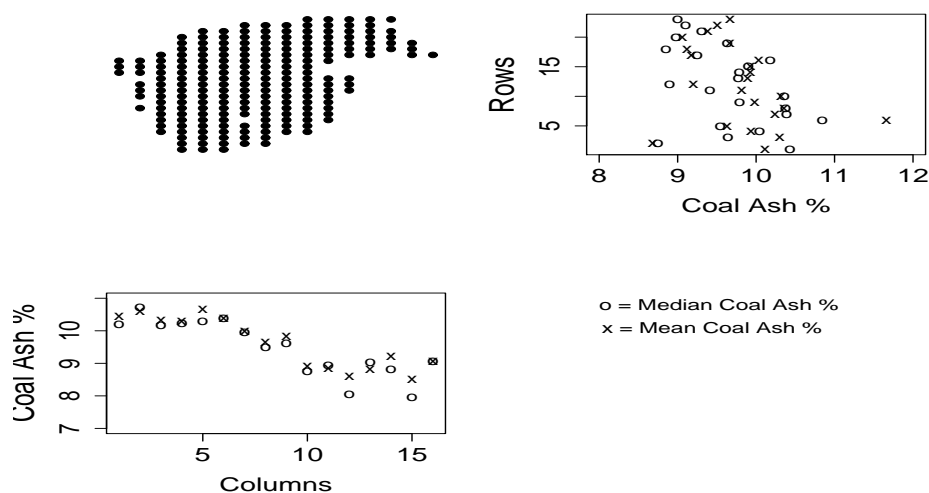
- This trend was also examined by considering boxplots for each of the rows and columns (shown below).



```
par(mfrow=c(2,1)) # Opens a 2x1 graphics window.
bwplot(y~coalash,data=coalash,subset=-50, # Creates a boxplot of the coal ash
      cex=1.3,xlab="Coal Ash %", ylab="y", # values row by row (y-value),
      main="Coal Ash % Row Summaries") # where the 50th value (outlier)
                                         # has been omitted from the plot.
bwplot(x~coalash,data=coalash,subset=-50, # Creates a boxplot of the coal ash
      cex=1.3,xlab="Coal Ash %", ylab="x", # values column by column (x-value),
      main="Coal Ash % Column Summaries") # omitting the outlier.
```

- And finally, the trend could also be seen by computing the means and medians for each row and column and displaying them as shown below.

Means and Medians Across Rows and Columns



Although there is clearly spatial trend across this area, it is not clear that the trend would easily be modeled by some function of the row and column coordinates. To perform a median polish of these data to remove the underlying trend, the following **R** commands were issued:

```

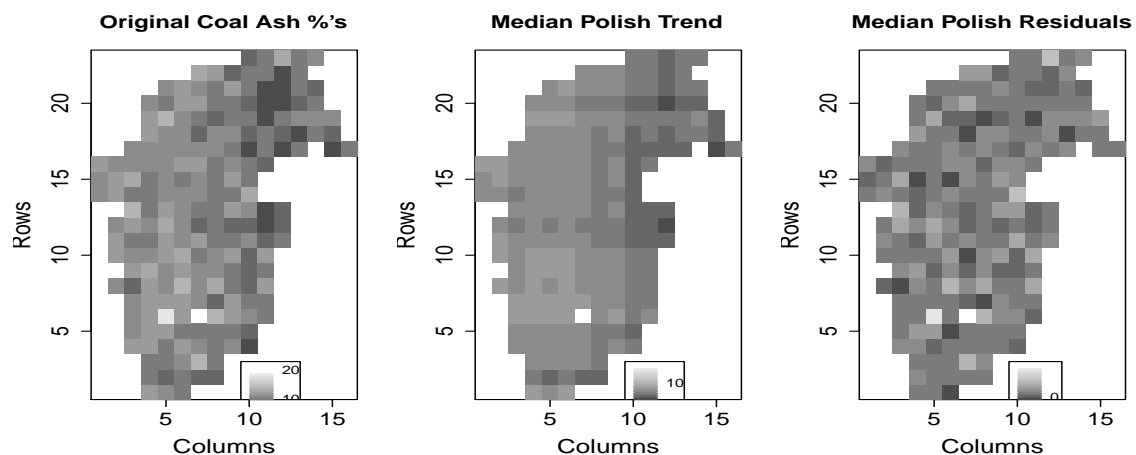
library(gstat)                                # Loads the "gstat" library.
data(coalash)                                # Selects coal ash data set from "gstat".
coal.mat <- tapply(coalash$coalash,           # Creates a matrix of the raw coal ash
  list(factor(coalash$x),factor              # percentages according to their
  (coalash$y)),function(x)x)                # locations.
coal.mp <- medpolish(coal.mat,                # Performs median polish on the coal ash
  na.rm=T)                                  # percentages.

```

- This function will return the grand median, vectors of row and column effects, and the resulting residuals under the names: grand, row, col, & residuals. To extract the trend or large-scale variation, we could issue the command:

```
coal.trend <- coal.mat - coal.mp$residuals
```

- One way to assess whether or not the median polish successfully removed any trend present is to compare greyscale plots of the regions before and after the polish. The resulting image plots are shown immediately below, with the corresponding **R** commands following.



```

par(mfrow=c(1,3))                            # Creates a 1x3 graphics window.
zmin <- min(coal.mat[!is.na(coal.mat)],        # Computes the minimum and maximum of
  coal.trend[!is.na(coal.trend)])              # all coal ash percentages and trend
zmax <- max(coal.mat[!is.na(coal.mat)],        # values, not including missing
  coal.trend[!is.na(coal.trend)])              # values (!is.na).
image(x=1:max(coalash$x),y=1:max(coalash$y),  # Creates a greyscale map of the coal
  coal.mat,zlim=c(zmin,zmax),cex.axis=1.5,    # ash %'s without interpolation.
  xlab="Columns",ylab="Rows",cex.lab=1.6,col=gray.colors(12))
image.legend(9.5,3,zlim=range(coal.mat,        # Puts legend on the plot with colors
  na.rm=T),col=gray.colors(12))                # matching those in image statement.
title("Original Coal Ash %'s",cex.main=1.5)    # Puts a title on the plot.
image(x=1:max(coalash$x),y=1:max(coalash$y),  # Creates a greyscale map of the coal
  coal.trend,zlim=c(zmin,zmax),cex.axis=1.5,  # ash trend values w/o interpolation.
  xlab="Columns",ylab="Rows",cex.lab=1.6,col=gray.colors(12))

```

```

image.legend(9.5,3,zlim=range(coal.trend,      # Puts legend on the plot with colors
na.rm=T),col=gray.colors(12))                # matching those in image statement.
title("Median Polish Trend",cex.main=1.5)      # Puts a title on the plot.
image(x=1:max(coalash$x),y=1:max(coalash$y), # Creates a greyscale map of the coal
coal.mp$resid,zlim=range(coal.mp$resid,      # ash residuals w/o interpolation.
na.rm=T),xlab="Columns",ylab="Rows",cex.lab=1.6,cex.axis=1.5,col=gray.colors(12))
image.legend(9.5,3,zlim=range(coal.mp$resid, # Puts legend on the plot with colors
na.rm=T),col=gray.colors(12))                # matching those in image statement.
title("Median Polish Residuals",cex.main=1.5)# Puts a title on the plot.

```

- One could also look at moving window statistics to see if the mean and variance of the median polish residuals are stable.

Final Comments on Median Polish

1. The technique of median polish is generally used prior to computing the variogram to remove any spatial trend present which is not well-described by some parsimonious function. After this large-scale variation is removed, all subsequent spatial analysis takes place on the median polish residuals. As estimation of the variogram presumes a stationary mean and variance, it is generally reasonable to assume that the residuals resulting from a median polish satisfy this property.
2. The median polish algorithm is not affected by missing data or irregular grids.
3. There are numerous other types of polishing that could be done. Instead of medians, one might choose to polish with means (if there are no gross outliers), trimmed means, or any type of weighted average of the data values. The basic decomposition of the trend into grand, row, and column effects will remain the same, and the algorithm will preserve this relationship throughout. Using medians is the most common choice due to their general robustness to erratic data values.
4. The median polish is not guaranteed to converge, but does in certain cases (see Fink, A.M. *How to Polish off Median Polish*. SIAM Journal of Scientific and Statistical Computing, 9, 932-940 (1988), for details). For this reason, a stopping rule is often defined for the algorithm of the form: Whenever none of the entries in the table change by more than ϵ at a given iteration, the algorithm is terminated.
5. Removal of the trend prior to modeling the covariance structure receives mixed reviews from users of spatial techniques. Some feel that removing the trend (i.e.: modeling the large-scale variation) introduces bias into the resulting residuals from which the covariance structure is extracted. This bias is a result of the choice of the trend function or method used. However, failure to remove any underlying trends negates the validity of the variogram due to the stationarity assumption. So, in either case, there is a problem. One final alternative is to model the trend and the covariance structure simultaneously, as is done with universal kriging. The difficulty here is that it is often very hard to separate which effects are due to large-scale and small-scale variation.