Lab03: Regression Kriging

Data Description

The zipped file **`KansasDEM.zip`** contains a DEM elevation points and the river network. UTM zone 14 is the underlying projection, so spherical distortions can be ignored in this example. The elevation (see ELEVATION) is measured in meters. Another variable (see RIVERDIST) measures the distance in miles from each grid cell to the nearest river. See the elevation map below.

Part I: Trend Surface Model (8 points)

Task 01: Use the script **`SampleByClick.r`** to pick manually in total 80 sample points. This task requires careful planning and perhaps a ***nested sampling strategy***. (2 points)

Clearly justify your selection strategies of sample points based on the criteria listed below: (2 points)

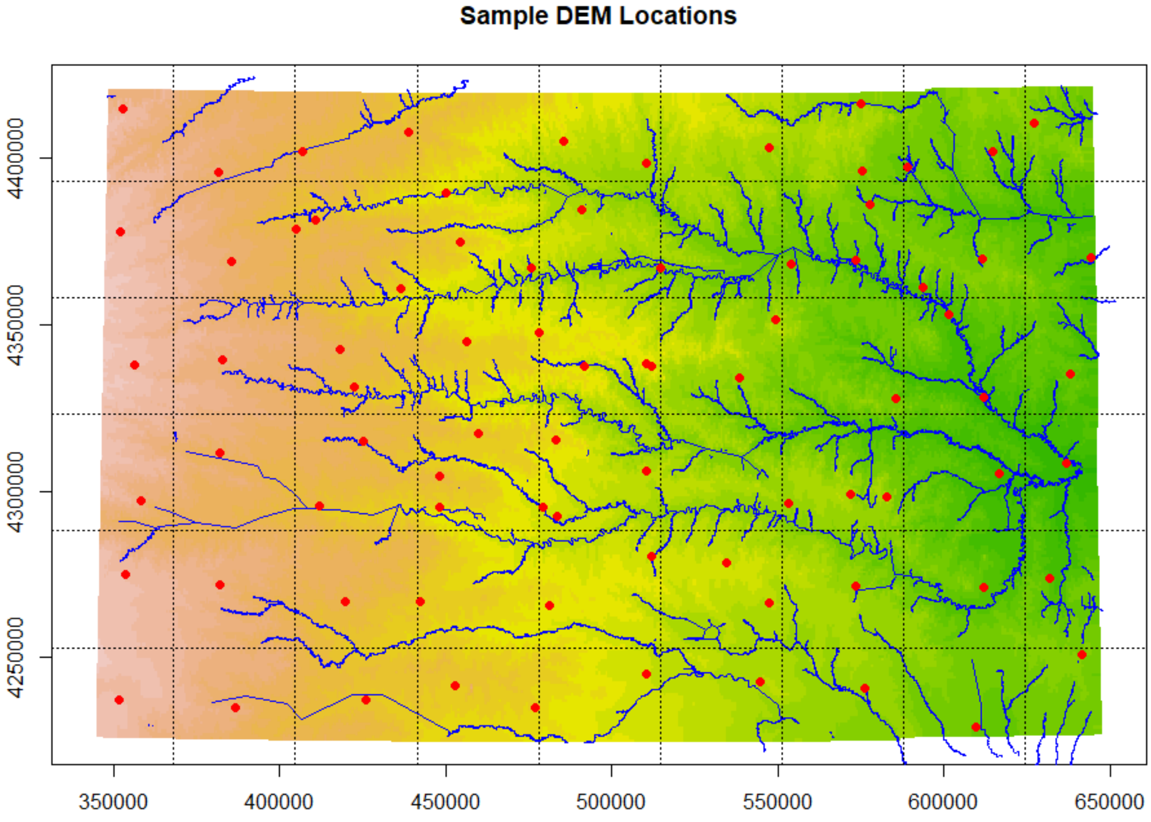


Figure 1. 80 sample points (denoted as red points) in DEM elevation map

(i) You want to avoid any bias in the predicted surface. Therefore, the average predicted elevations should match closely the average observed elevations in the study area. *How can you try to avoid this potential bias?*

Answer: There are 4 steps in my sampling scheme. **Step 1:** Plot a grid in the study area. Place an 6x9 grid in the plot based on the bounding box of the DEM elevation map. Within bounding of elevation map, randomly place one sample point within each grid. Therefore, there are 54 points evenly placed over the elevation map. **Step 2:** Assign some points in the edge of study area. **Step 3:** Place 10 points in the area with rapid change of elevation between ridge and river. Calculate the mean elevation of 70 sample points. **Step 4:** Based on the mean elevation of 70 sample points, place the remaining 10 points to get the mean sample elevation as close as possible to the mean DEM elevation.

(ii) The **extrapolation** problem should be avoided and the prediction error, in particular at the edges of the study area, need to be minimized. *How many sample points should be assigned to control for this problem?*

Answer: In my sampling scheme, the extrapolation problem has been avoided. First, along the edge of the elevation map, there are 26 grids, which ensures that on average about 10 points will fall close to the edge of the map. Then, 6 points will be used to adjust the distribution of points along the edge. Therefore, the edge problem has been fully considered and addressed.

(iii) The rapid topographic variation along the river valleys and ridges needs to be captured properly. *How many sample points should be assigned to model this variability, where should they be placed and which variable in the data set measures it?*

Answer: I have placed 10 points to capture this rapid elevation change. Some of the points are close to the river, some are on the ridges. The variable of ELEVATION of those 10 sample points will be used to measure the variation. Combining those 10 sample points with the previous 60 points, we calculated the mean sample elevation—605.157, which is close to the mean elevation.

```
> samplePts70 <- spRbind(samplePts60,samplePts10.1)

> mean(samplePts70$ELEVATION)

[1] 605.1571

> mean(grid.data$ELEVATION)

[1] 603.2856
```

(iv) In order to build a well-defined variogram all spatial scales of the inter-sample point distances need to be represented. *How many sample points should be assign to fill in missing distance ranges and where should these points be placed?*

Answer: In order to build a well-defined variogram, the inter-sample point distance should be in variety. Since there are very few points clustered together, we put some points close enough to have a small inter-sample point distance. After the 80 sample points are placed, we check the mean sample elevation—599.1625, still close to the DEM mean elevation 603.2856. The 80 sampling points are shown in figure 1.

```
> samplePts80 <- spRbind(samplePts70,samplePts10.2)

> mean(samplePts80$ELEVATION)

[1] 599.1625
```

Task 02: Estimate the 1st, 2nd and 3rd order trend-surface models. Include the distance to the nearest rivers as covariable. (1 point)

```
## 1st Order Trendsurface
polyForm1<makeTrendPolyForm(ELEVATION~RIVERDIST,~caliX+caliY,polyDeg=1)
polyForm1
trendsurf.1<- lm(polyForm1,data = spl80)
summary(trendsurf.1)

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)           585.875      6.211  94.328  < 2e-16 ***
RIVERDIST               7.159      1.936   3.697  0.00041 ***
I(caliX^1 * caliY^0) -126.895      4.329 -29.311  < 2e-16 ***
I(caliX^0 * caliY^1)   -8.019      4.132  -1.941  0.05598 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35.3 on 76 degrees of freedom
Multiple R-squared:  0.9388,  Adjusted R-squared:  0.9364 
F-statistic: 388.8 on 3 and 76 DF,  p-value: < 2.2e-16
```

```
## 2nd Order Trendsurface
polyForm2<makeTrendPolyForm(ELEVATION~RIVERDIST,~caliX+caliY,polyDeg=2)
polyForm2
trendsurf.2<- lm(polyForm2,data = spl80)
summary(trendsurf.2)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)           563.5967     6.9643  80.927  < 2e-16 ***
RIVERDIST               6.0205     1.6711   3.603 0.000572 ***
I(caliX^1 * caliY^0) -128.3194     3.7293 -34.408  < 2e-16 ***
I(caliX^0 * caliY^1)   -8.7696     3.5429  -2.475 0.015636 *  
I(caliX^2 * caliY^0)   18.9621     3.7033   5.120 2.39e-06 ***
I(caliX^1 * caliY^1)   -0.4753     3.3741  -0.141 0.888351    
I(caliX^0 * caliY^2)    6.5884     3.9427   1.671 0.098992 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 30.21 on 73 degrees of freedom
Multiple R-squared:  0.957,   Adjusted R-squared:  0.9534 
F-statistic: 270.6 on 6 and 73 DF,  p-value: < 2.2e-16
```

```
## 3rd Order Trendsurface
polyForm3<makeTrendPolyForm(ELEVATION~RIVERDIST,~caliX+caliY,polyDeg=3)
polyForm3
trendsurf.3<- lm(polyForm3,data = spl80)
summary(trendsurf.3)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)           564.3283     6.7570  83.518  < 2e-16 ***
RIVERDIST               5.8081     1.6266   3.571 0.000654 ***
I(caliX^1 * caliY^0) -134.5585     8.6100 -15.628  < 2e-16 ***
I(caliX^0 * caliY^1)  -28.4476     8.7821  -3.239 0.001845 ** 
I(caliX^2 * caliY^0)   19.3966     3.5708   5.432 7.85e-07 ***
I(caliX^1 * caliY^1)   -0.7181     3.2530  -0.221 0.825933    
I(caliX^0 * caliY^2)    6.2139     3.7928   1.638 0.105902    
I(caliX^3 * caliY^0)   -0.9471     3.9442  -0.240 0.810940    
I(caliX^2 * caliY^1)    2.7838     3.6451   0.764 0.447640    
I(caliX^1 * caliY^2)    7.5505     3.6176   2.087 0.040570 *  
I(caliX^0 * caliY^3)    9.3916     4.4288   2.121 0.037554 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 28.95 on 69 degrees of freedom
Multiple R-squared:  0.9627,  Adjusted R-squared:  0.9572
F-statistic: 177.9 on 10 and 69 DF,  p-value: < 2.2e-16
```

Task 03: Map the three predicted trend-surfaces. Use a meaningful color ramp. (1 point)

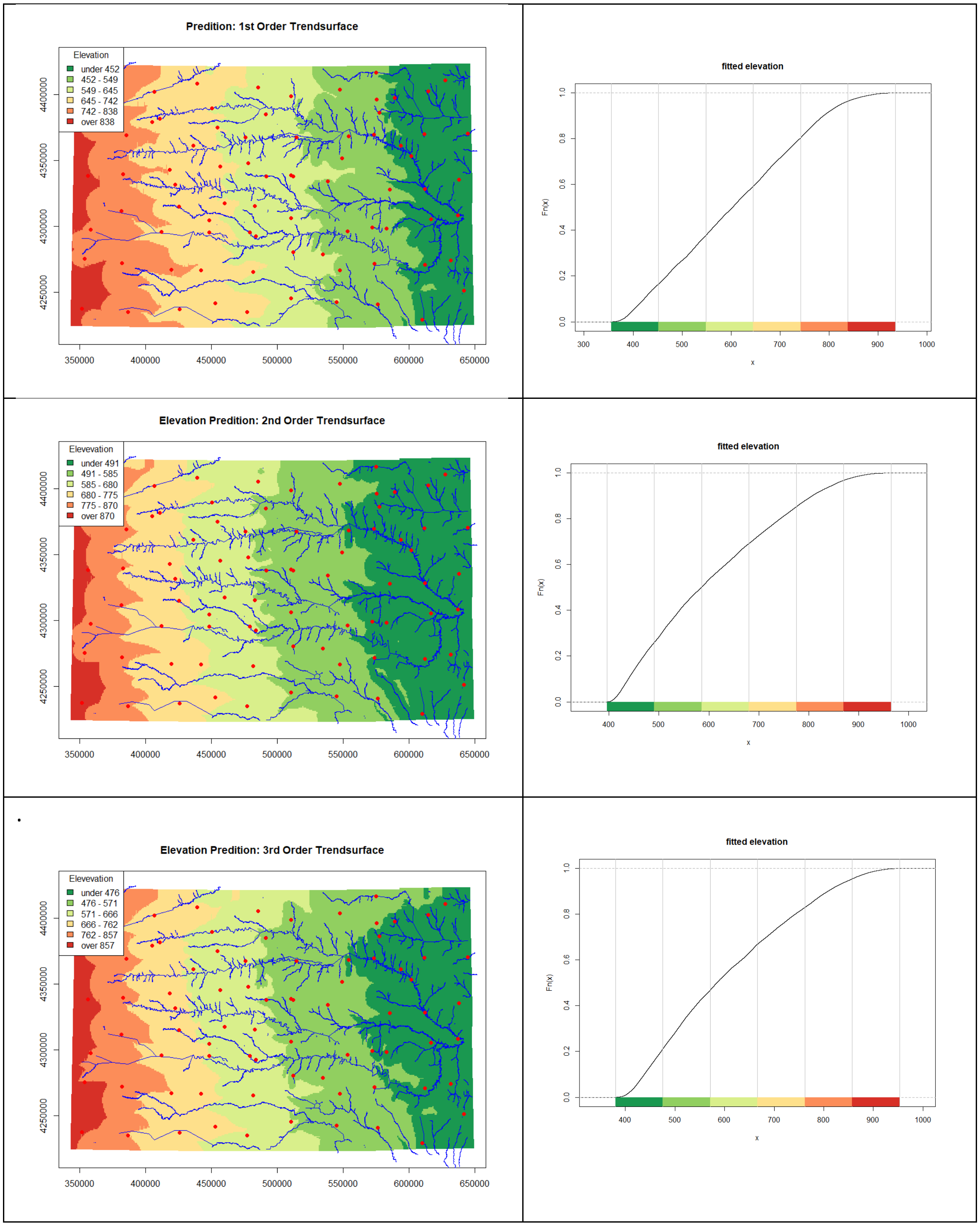


Figure 2. 1st, 2nd, 3rd order Trend surface and corresponding elevation classification scheme

Task 04: Decide with the partial *F*-test, which of the three surface models is most appropriate for your given sample points. Interpret the selected trend-surface regression model. (1 point)

```
> ## compare 1st order with 2nd order Trendsurface
> anova(trendsurf.1,trendsurf.2)
Analysis of Variance Table

Model 1: ELEVATION ~ RIVERDIST + I(caliX^1 * caliY^0) + I(caliX^0 * caliY^1)
Model 2: ELEVATION ~ RIVERDIST + I(caliX^1 * caliY^0) + I(caliX^0 * caliY^1) + 
    I(caliX^2 * caliY^0) + I(caliX^1 * caliY^1) + I(caliX^0 * 
    caliY^2)
  Res.Df   RSS Df Sum of Sq     F    Pr(>F)    
1     76 94715                                 
2     73 66624  3     28091 10.26 1.024e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> ## compare 2nd order with 3rd order Trendsurface
> anova(trendsurf.2,trendsurf.3)
Analysis of Variance Table

Model 1: ELEVATION ~ RIVERDIST + I(caliX^1 * caliY^0) + I(caliX^0 * caliY^1) + 
    I(caliX^2 * caliY^0) + I(caliX^1 * caliY^1) + I(caliX^0 * 
    caliY^2)
Model 2: ELEVATION ~ RIVERDIST + I(caliX^1 * caliY^0) + I(caliX^0 * caliY^1) + 
    I(caliX^2 * caliY^0) + I(caliX^1 * caliY^1) + I(caliX^0 * 
    caliY^2) + I(caliX^3 * caliY^0) + I(caliX^2 * caliY^1) + 
    I(caliX^1 * caliY^2) + I(caliX^0 * caliY^3)
  Res.Df   RSS Df Sum of Sq      F  Pr(>F)  
1     73 66624                              
2     69 57817  4    8807.8 2.6279 0.04175 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Answer: Based on my sample points, the nested partial F-tests suggest that 2nd order trend surface model is better than 1st order at a significant level of 0, and 3rd order trend surface model is better than 2nd order at a significant level of 0.05. Therefore, I think the 3rd order trend surface model is the best one to predict the DEM elevation map.

The output of 3rd order trend surface model for the 80 sample points tells us that 95.72% variation of elevation has been explained by this model. The variable of RIVERDIST --distance from the nearest river is positively correlated with the value of elevation, which means that the elevation tends to be lower when closer to the river. The X, Y coordinates also significantly explain the change of elevation, however there exists non-linear relationship. The overall pattern of elevation with the coordinates are: as X coordinates tend to be larger from West to East, elevation become smaller; with the increase of Y coordinates from South to North, elevation tender to be smaller.

Task 05: Evaluate the prediction quality of your most appropriate trend surface model. Does the histogram of observe elevations match that based on the predicted values? Does your prediction model lead to biased overall elevation estimates? If yes, what may be the cause? (1 point)

```
## task 5--histogram
breakpts <- seq(250,1000,by=25)          # Define Elevation classes
## prediction
hist(predTrend3$fit, breaks=breakpts,xlab="Elevation(meters)",
     ylim = range(c(0,9000)),main="Distribution of 3rd Order Prediction Elevations")
## original data
hist(gridDEM$ELEVATION,breaks=breakpts,xlab="Elevation(meters)",
     ylim = range(c(0,9000)),main="Distribution of DEM Elevations")
```

```
summary(cbind(predTrend3$fit, gridDEM$ELEVATION))
e1071::skewness(predTrend3$fit);e1071::skewness(gridDEM$ELEVATION)
e1071::kurtosis(predTrend3$fit);e1071::kurtosis(gridDEM$ELEVATION)
```

Answer: the overall pattern of prediction histogram does match the observed ones. Both of two histograms are **unimodal**, and **peaks** between elevations of 400-500 meters. Besides, both two histograms are **positively skewed**. However, there exists some bias in the prediction model. Firstly, much local variation of the elevation has been removed, and the right tail of the prediction histogram is longer than the actual DEM elevation histogram. The reason behind this is that those area with rapid elevation change and with very low and very high elevation has been **smoothed** in the trend surface model.

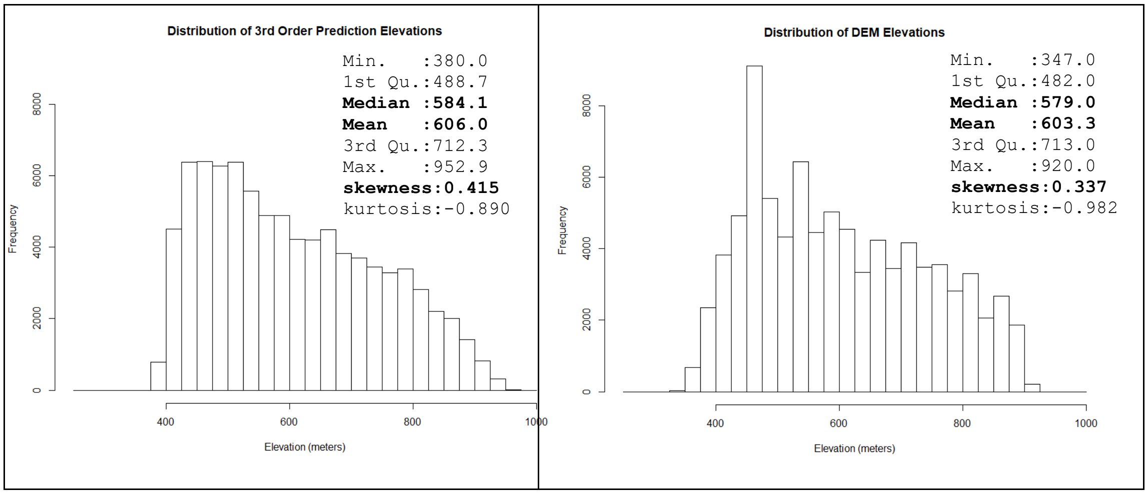


Figure 3. prediction elevation histogram(left) vs. DEM elevation histogram(right)

Task 06: For your most appropriate model, map the standard errors of the prediction surfaces. Use a meaningful color ramp. Interpret the general pattern in the standard errors. In particular evaluate the standard errors at the edges of the study area relative to those in the center? (1 point)

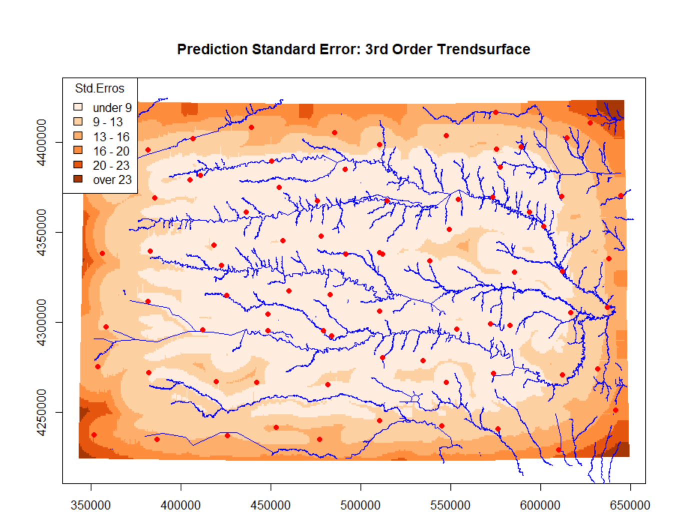


Figure 4. Blue lines are river network; red dots are 80 sample points

General pattern: The central part of the study area has the smallest standard errors. With the distance closer to the edges of the study area, standard errors tend to be larger, and the four corners of the study area has the largest standard errors.

Overall, the standard errors have been minimized with the evenly distributed sample points, and the areas with large standard errors in the four corners are very small because of those sample points near the edges.

```
## task 6--Map standard error
## Map 3rd Order Prediction Uncertainty
predSe.3 <- predTrend3$se
n.col <- 6
pal <- brewer.pal(n.col,"Oranges")
seClass <- classIntervals(predSe.3, n.col, style="equal")

seCol <- findColours(seClass,pal)
plot(gridDEM,axes=T,col=seCol,pch=15,cex=2,
     main="Prediction Standard Error: 3rd Order Trendsurface")
plot(river,col="blue",add=T)
plot(samplePts80,add=T, col="red", pch=19)
legend("topleft", title = "Std.Erros",
       legend = leglabs(round(seClass$brks, digits = 0)), fill = pal, bty = "o", ncol = 1)
```

Task 07: For your most appropriate model, calculate the error component (residual surface: observed DEM minus predicted trend DEM). Map this pattern with a bipolar map theme (zero is the neutral value) and overlay the river network onto your residual map. Interpret this residual pattern. (1 point)

```
## task 7-- map 3rd order residual surface
pred.res <- gridDEM$ELEVATION-predTrend3$fit
n.col <- 5
pal <- rev(brewer.pal(n.col,"RdBu"))
seClass <- classIntervals(pred.res, n.col, style="equal")
seClass
seCol <- findColours(seClass,pal)
plot(gridDEM,axes=T,col=seCol,pch=15,cex=2,
     main="3rd Order Trend residual surface")
plot(river,col="blue",add=T)
legend("topleft", title = "Residual",
       legend = leglabs(round(seClass$brks, digits = 0)), fill = pal,
       bty = "o", ncol = 1)
```

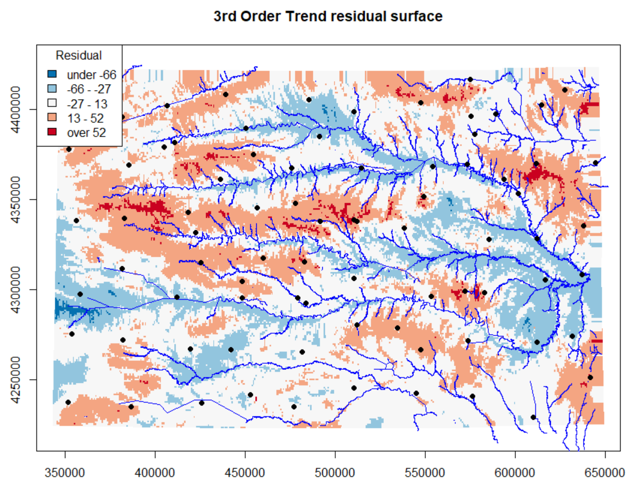


Figure 5. Blue lines -- river network; black dots-- 80 sample points

The residual mostly lies in the area with rapid elevation change between river and mountain ridge. The pattern of the residual is that: along the river network, there are negative residuals, indicating an overestimation--prediction is higher than actual elevation; along the mountain ridges, positive residuals are widely distributed because of the underestimation from the 3rd order trend surface model -- predicted elevation is lower than actual one.

Part II: Variogram Estimation (2 points)

Task 08: Estimate the variogram function based on the error component at the sampling locations from Part I. Show the necessary plots and interpret them by exploring possible anisotropy, range, sill and nugget effects.

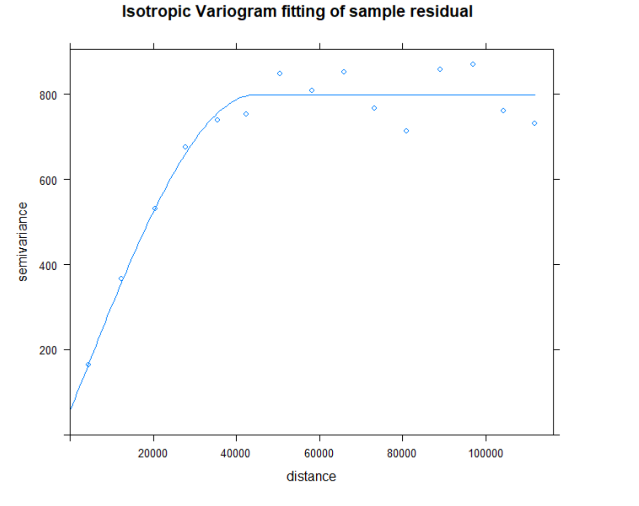
```
## task 8—use 80 sample points residual to fit variagram
samplePts80$resid<-trendsurf.3$residuals
```

```
## isotropic variogram
Library(gstat)
eleiso.vgm <- variogram(resid~1, data = samplePts80,cloud= F)
eleiso.vgm
plot(eleiso.vgm)

## isotropic fitting
eleiso.fit <- fit.variogram(eleiso.vgm, model=vgm(psill=6000,model="Exp",range=50000, nugget=100))
eleiso.fit
plot(eleiso.vgm, model=eleiso.fit, as.table=TRUE,
     main="Isotropic Variogram fitting of sample residual")

## anisotropic variogram
eleAniso.vgm <- variogram(resid~1, data = samplePts80,alpha=c(0,45,90,135))
plot(eleAniso.vgm)
plot(variogram(resid~1, data = samplePts80, map=TRUE, cutoff=40000, width=600),
     main="Radar Map to test anisotrophy")  ##-- radar map

## anisotropic fitting
eleAniso.fit<-fit.variogram(eleAniso.vgm, model=vgm(psill=6000,model="Exp",range=50000,nugget=500))
plot(eleAniso.vgm, model=eleAniso.fit,as.table=TRUE, main= "Directional variogram plots")
```

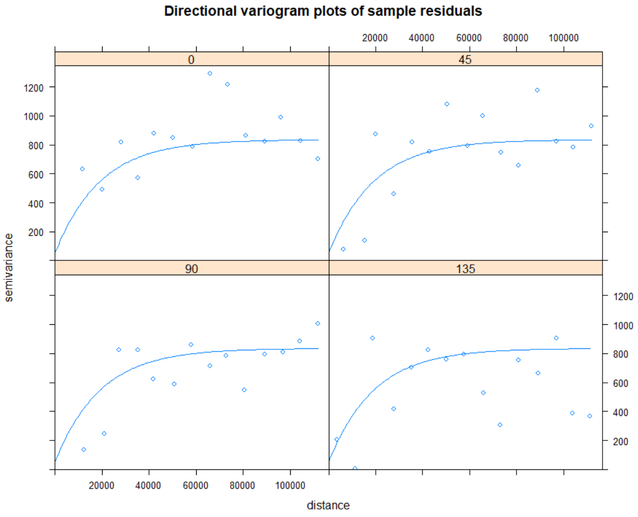


```
> eleiso.fit
  model      psill   range
1   Nug   56.97381     0.0
2   Sph  741.67029 44452.7

> var(samplePts80$resid)
[1] 731.8552
```

The fitted variogram has a **nugget** effect of 56.97, **sill** value of 741.67, and **range** of 44452.7 meter. The fitted variogram increases rapidly with distance below 44452.7 meter, and reaches the sill at the range distance, which means no spatial autocorrelation above distance of 44452.7 meter between points. The sill value is very close to the variance of the sample points.

Both the Anisotropic variograms and Radar map show no anisotropy pattern in the sample residuals.



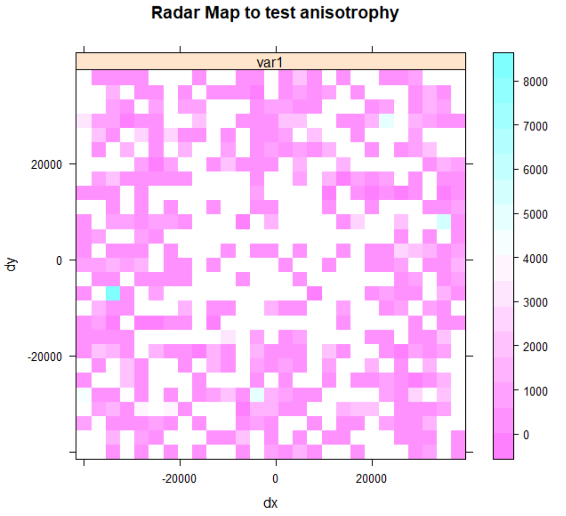


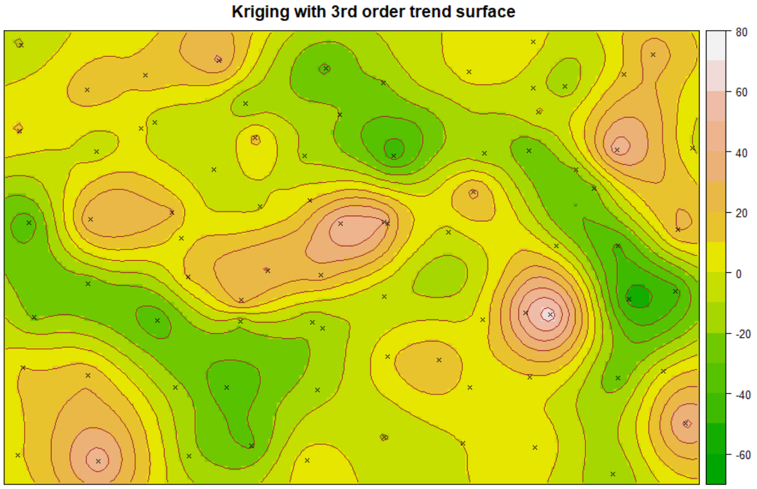
Figure 6. Sample points elevation residual variogram plots

Part III: Kriging Interpolation of the error component (3 points)

Task 09: Predict the error component by Kriging for all locations. Justify your choice of the Kriging model. Map the surface of the predicted error component with an appropriate color ramp. (1.5 point)

```
## Predict the error component by Kriging
## project the sample points
proj4string(samplePts80) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84
                                 +towgs84=0,0,0")
ele.gstat <- gstat(id="simKrig", formula=resid ~1, beta= 0, data=samplePts80)
ele.gstat <- gstat(ele.gstat, id="simKrig", model=eleiso.fit)           # augment object
ele.pred <- predict(ele.gstat, newdata= gridDEM)

## Plot Predicted Kring elevation
pts <- list("sp.points", samplePts80, pch=4, col="black", cex=0.5)
spplot(ele.pred, zcol="simKrig.pred", col.regions=terrain.colors(20), cuts=19,
       sp.layout=list(pts), contour=T, labels=FALSE, pretty=TRUE, col="brown",
       main="Kriging with 3rd order trend surface")
```



Answer: Simple Kriging was used in my plot, because the mean value of the residual surface is known. After the 1st component has been removed in the dataset by 3rd order trend surface, the mean value of residuals is 0.

Figure 7. error component prediction

Task 10: Estimate the uncertainty of the error component for all locations. Map the uncertainty surface with an appropriate color ramp. (1.5 point)

```
## Plot Predicted Uncertainty surface
spplot(ele.pred,zcol="simKrig.var",
      col.regions=rev(heat.colors(10)),
      cuts=9,sp.layout=list(pts),
      main="Uncertainty        Surface:
Prediction Variance")
```

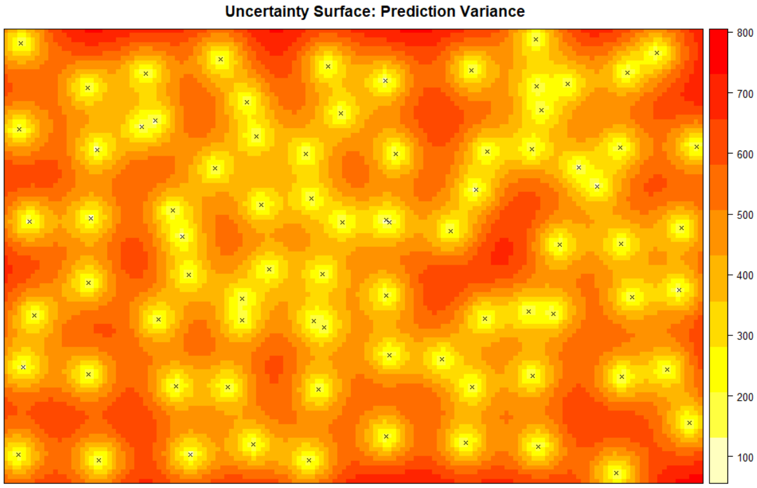


Figure 8. error component prediction uncertainty

Part IV: Combining Frist and Second Order Components (3 points)

Task 11: Combine the predicted trend-surface with the predicted error component to obtain the overall predicted DEM surface. Map this predicted surface with a proper color ramp. (1 point)

```
# combine trend surface prediction with kriging prediction
gridDEM$pred.1comb2 <- predTrend3$fit +ele.pred$simKrig.pred

# plot the surface
breakpts <- seq(250,1000,by=25)
ncl <- length(breakpts)-1
pal <- terrain.colors(ncl)
cols <- pal[findInterval(gridDEM$pred.1comb2,breakpts,rightmost.closed=T)]
plot(gridDEM,axes=T,col=cols,pch=15,cex=1)
plot(river,col="blue",add=T)
plot(samplePts80,add=T, col="red", pch=19)
title("Elevation prediction: 1st order, 2nd order components combined")
```

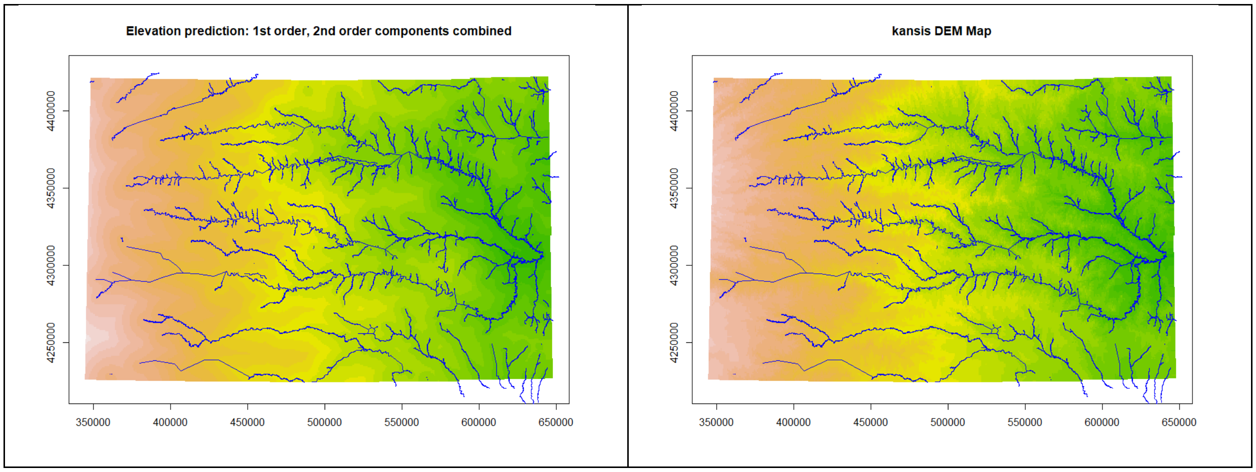


Figure 9. Left map is elevation prediction map combined the 1st and 2nd order components; right map is the observed DEM map. For two maps comparable, both maps are in the same elevation classification color scheme.

Comparing two maps, we can conclude that a very good prediction was obtained by combining 1st and 2nd order components. However, there is still some bias in the prediction map, which mostly comes from the area where there is a rapid elevation change between river and ridge. The prediction map captured the change, however, smoothed the variation.

Task 12: Combined the trend-surface **prediction uncertainty** with the **kriging uncertainty** in the standard deviation scale. Map the uncertainty surface with a proper color ramp. (1 point)

```
## calcualte the combined standard error from 1st component and 2nd component
gridDEM$comb.sd <- sqrt((predTrend3$se.fit)^2 + ele.pred$simKrig.var)

n.col <- 6
pal <- brewer.pal(n.col,"Oranges")
seClass <- classInt::classIntervals(gridDEM$comb.sd, n.col, style="equal")
seCol <- classInt::findColours(seClass,pal)
plot(gridDEM,axes=T,col=seCol,pch=15,cex=2,
     main="prediction uncertainty in standard deviation scale:1st and 2nd component combined")
plot(river,col="blue",add=T)
plot(samplePts80,add=T, col="black", pch=19)
```

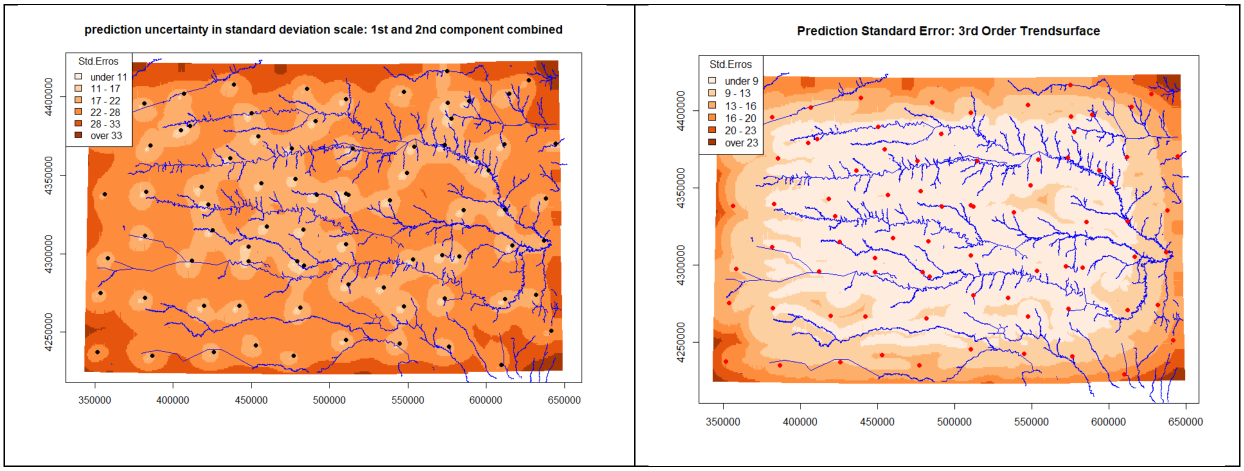


Figure 10. left is total Prediction Uncertainty map combining 1st and 2nd component; right is just 1st component prediction uncertainty map from task 6 for comparison.

Task 13: Calculate the **root mean squared error** of your overall predicted DEM values by comparing it against the observed DEM value of the Kansas topographic surface. (1 point)

```
> # calcualte the residual from 1st and 2nd fit
> predErr <- gridDEM$ELEVATION-gridDEM$pred.1comb2
> # calculate total RMSE from overall prediction
> (RMSE <- sqrt(sum(predErr^2)/length(gridDEM$pred.1comb2)))
[1] 21.45598

# map the final residuals
n.col <- 5
pal <- rev(brewer.pal(n.col,"RdBu"))
seClass <- classIntervals(pred.res, n.col, style="equal")
seCol <- findColours(seClass,pal)
plot(gridDEM,axes=T,col=seCol,pch=15,cex=2,
     main="3rd Order Trend residual surface")
plot(river,col="blue",add=T)
legend("topleft", title = "Residual",
       legend = leglabs(round(seClass$brks, digits = 0)), fill = pal, bty = "o", ncol = 1)
```

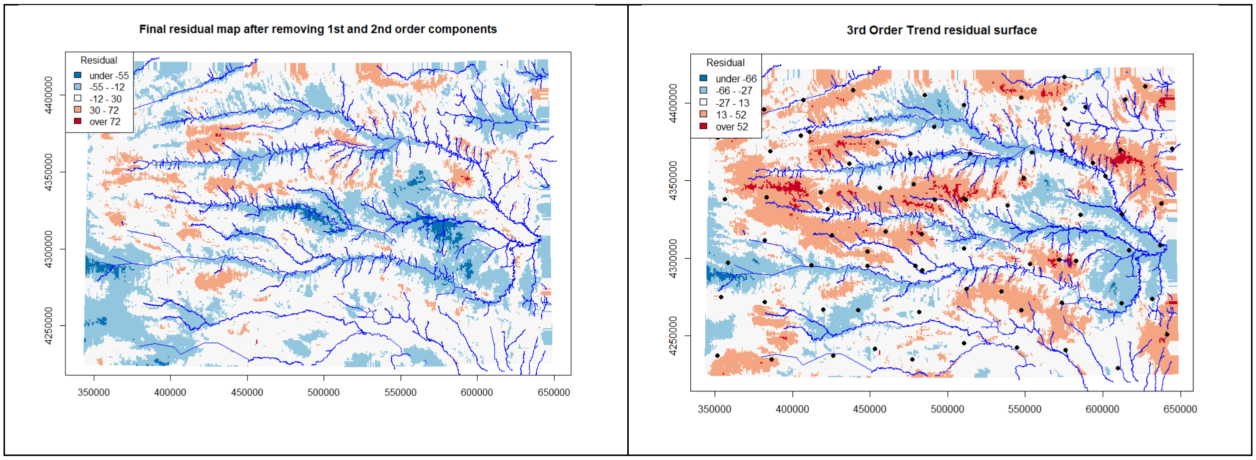


Figure 11. left is total prediction (1st order fit and 2nd order fit) residual map; right is just trend surface residual map from task 7 for comparison.

Comparing above 2 maps, we can easily see that the overall prediction residuals has been significantly reduced by adding the second component fit through kriging prediction, which considers about the spatial autocorrelation between points.