Exam 1 answers

Philip Dixon

4/8/2020

Range of scores: 55 - 99 out of 100

Average score: 81

Median score: 83

Very, very rough conversion to letter grades:

> 80 some form of A < 85 some form of B

Yes, these overlap. The overlap area reflects "I don't know, at least right now.

Frequently missed questions: 1a, 1c, 2a, 4, 6b.

General comments about the data analysis:

Some folks didn't follow my requested format, which cost a few points. If you haven't looked at a scientific paper, you may not know how a methods section is structured. I have posted two nice examples of methods paragraphs. Both have been lightly edited. One is short (about as short as it could be); the other is longer.

I deducted 3 points if you didn't check any assumptions (isotropy, no apparent outliers, constant variance). To get credit, you just had to check one. If these data mattered to you, you should check all.

Short answer questions.

4 pts each part, with partial credit.

1) Sampling yellow perch

a) No - you can't use information from one sample to ascertain bias.

Notes: Bias is a comparison of the average estimate to the true value. You need more than one sample to estimate the average. One very different sample average could be bias (consistently bad) or just large variability (this sample was bad). This was a commonly missed point.

b) Yes, methods A and B are the least biased. The mean values in the table are closest to the true value. Note: You could argue that all methods are unbiased because the 95% confidence intervals for each method mean includes the true value.
c) Method E is the most precise. It has the smallest standard deviation. Note: the standard error only tells you how well the mean has been estimated. Doesn't tell you about the variability between estimates from different samples.

2) Singular model fits

a) The variogram data can not separate the measurement error variance and the nugget variance. All it can do is estimate the total. Note: This is why we usually specify the meas. error variance. This would be calculated separately from repeat measurements of the same samples.
b) There isn't data at enough distances to estimate a nugget and range. The four points at larger distances estimate the sill. The other two parameters are estimated from 1 data point. Need more semi-variance bins at small distances.

3) Kriging concepts:

a) 2.0. The neighbors of P1 are within the practical range and their values are 1.5 and 2.7.

b) 10. P2 has no neighbors within the practical range, so the prediction will be the GLS mean.

c) 10. P1 is close to neighbors, so expect variance $< \sigma^2 = 34.6$

d) 38.0 The variance of P2 has two components: $\sigma^2 = 34.6$ (because the prediction far from neighbors) and the variance of the GLS mean (but you don't know how bit this is).
Note: 34.6 with an appropriate explanation was accepted for full credit.

4) Residuals - example 1. Nothing - all the residuals are zero, which you expect because you are predicting at locations with data. Note: The prediction is the observed value, so the residual = Y - prediction = 0.

5) Residuals - example 2. Yes, I'm concerned about unequal variance. The residual plot shows a trumpet shape characteristic of unequal variance.

6) Distances

a) 3 - very similar distances. The points are close so the distance is short.
b) 1 - need to work within a single UTM zone.
c) 2 - NAD83 and WGS84 have been reconciled, so can locations are similar.

Data analysis problem:

There are many reasonable answers to this problem, because there are many decisions where one choice is clearly right and others are clearly wrong.

You didn't need the outline of the northern Gulf of Mexico. Both grids were trimmed to the GOM.

My answers: a) Everybody's text will be different.
I was looking for: universal kriging with a linear trend model, which variogram estimator, criterion for choosing a variogram model, your choice of variogram model.

Nice things to include (at least concisely) are that you evaluated assumptions (constant variance, isotropy) and what software you used.

Two examples are given in methodsText.docx.

b) Some of the decisions:

- choice of variogram estimator
- starting values for variogram modeling
- choice of variogram model
- evaluation of at least one assumption (isotropy, equal variance)

c) My results were:

coordinates PredSST sd

(-91.1, 28.6) 18.33 0.51

(-86.2, 26.35) 25.90 0.24

(-87.6, 24.6) 26.40 0.42

(-85.1, 24) 26.52 0.59

d) And my fine grid is at the bottom of the code

My R code.

Load needed packages

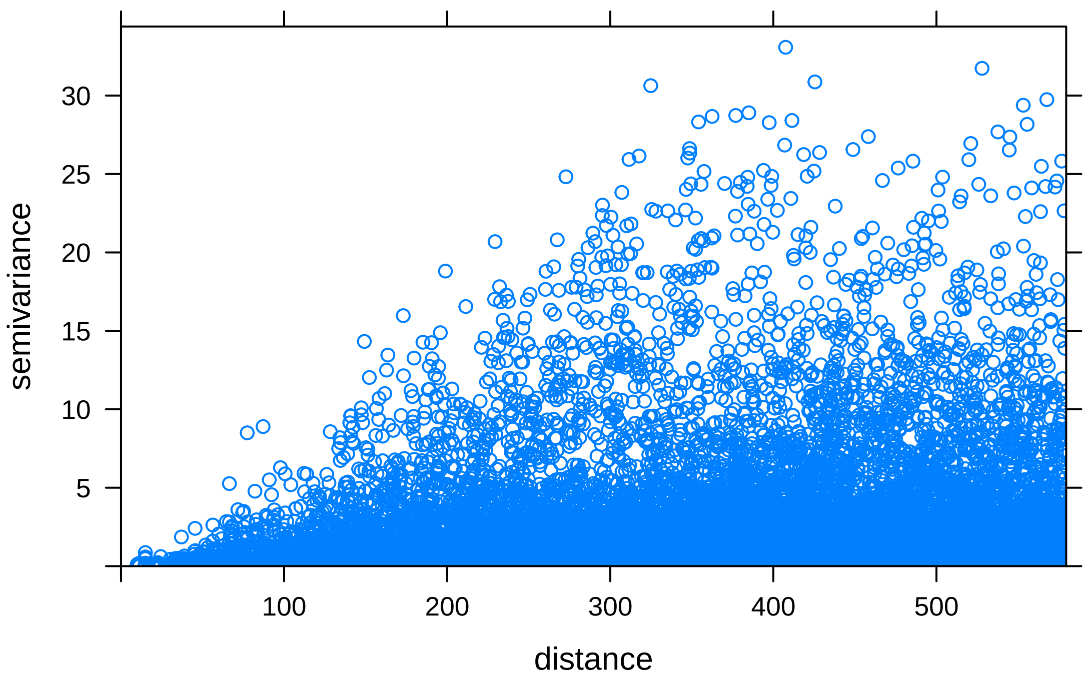
```
library(sp)
library(gstat)
```

Load the data and convert to spatial objects

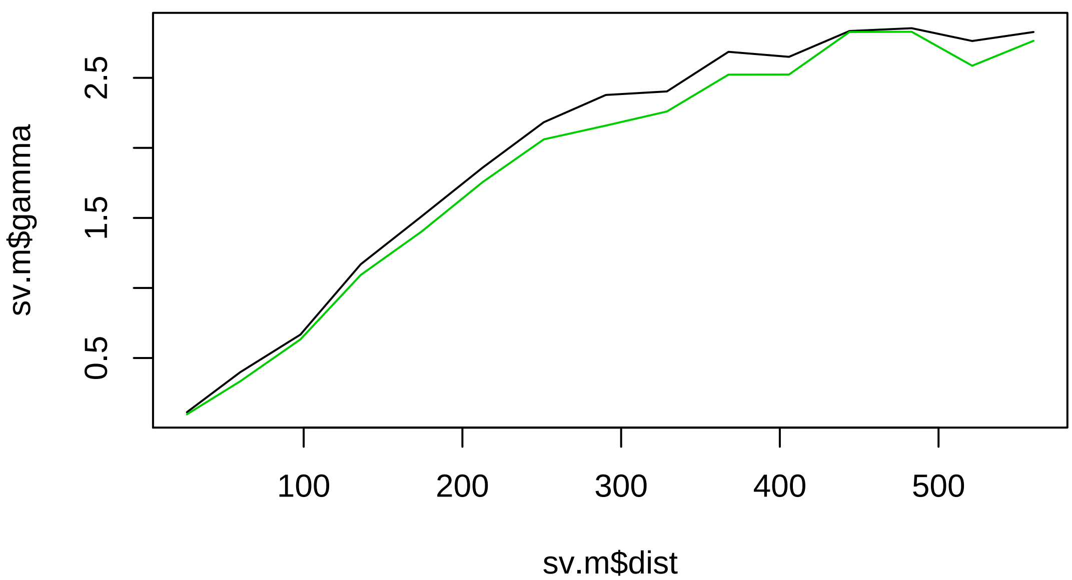
```
sst <- read.csv('sstdata.csv', as.is=T)
finegrid <- read.csv('sstgrid.csv', as.is=T)
grid <-  read.csv('sstgrid2.csv', as.is=T)
sstoutline <- read.csv('sstoutline.csv', as.is=T)

sst.sp <- sst
coordinates(sst.sp) <- c('long','lat')
proj4string(sst.sp) <- CRS('+proj=longlat +datum=WGS84')

finegrid.sp <- finegrid
coordinates(finegrid.sp) <- c('long','lat')
proj4string(finegrid.sp) <- CRS('+proj=longlat +datum=WGS84')
gridded(finegrid.sp) <- T

grid.sp <- grid
coordinates(grid.sp) <- c('long','lat')
proj4string(grid.sp) <- CRS('+proj=longlat +datum=WGS84')
gridded(grid.sp) <- T

outline.sp <- sstoutline
coordinates(outline.sp) <- c('long','lat')
proj4string(outline.sp) <- CRS('+proj=longlat +datum=WGS84')
```

The Matheron semivariogram:

```
plot(variogram(sst ~ long + lat, data=sst.sp, cloud=T) )
```



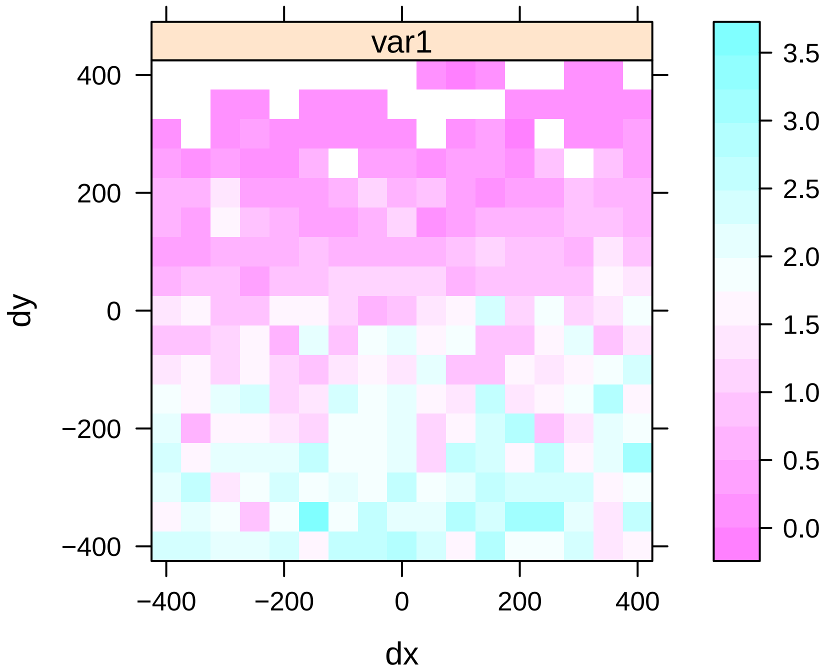
```
sv.m <- variogram(sst ~ long + lat, data=sst.sp)
sv.ch <- variogram(sst ~ long + lat, data=sst.sp, cressie=T)

# to overlay the Matheron and CH variograms
plot(sv.m$dist, sv.m$gamma, type='l')
lines(sv.ch$dist, sv.ch$gamma, col=3)
```



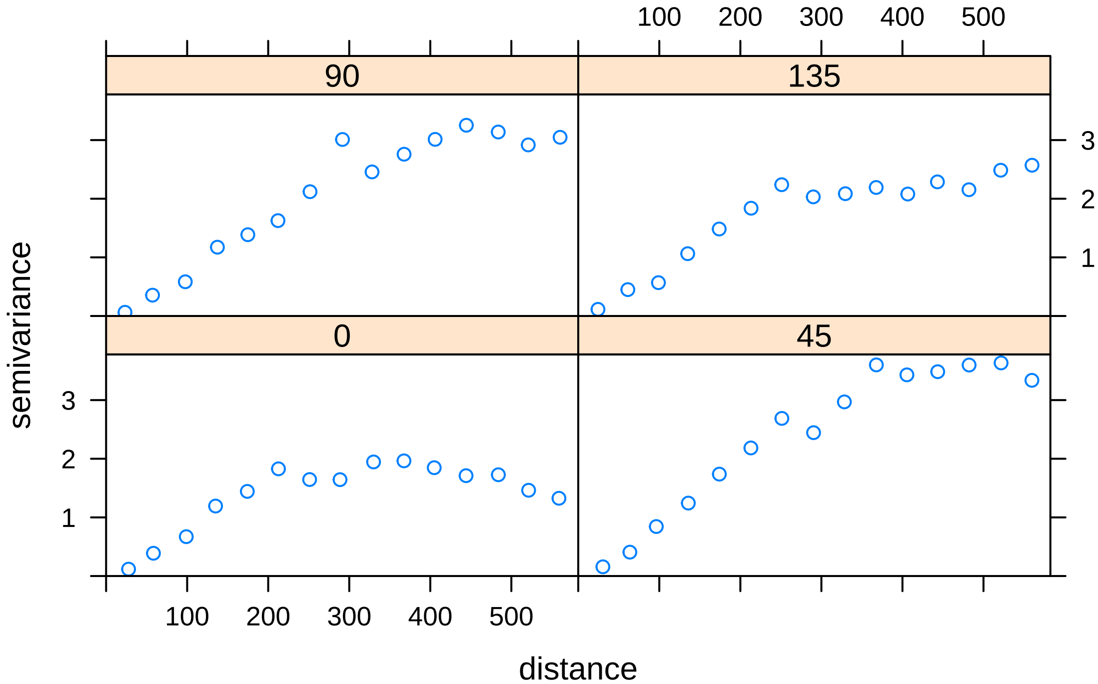
The variogram cloud doesn't seem to have any outliers. The Cressie-Hawkins estimator is about the same as the Matheron. The default parameters provide lots of distances before you get to the sill.

Examine anisotropy

```
plot(variogram(sst ~ long + lat, data=sst.sp, map=T,
  cutoff=400, width=50) )
```




```
sv.anis <- variogram(sst ~ long + lat, data=sst.sp,
  alpha=c(0, 45, 90, 135), tol.hor=22.5)
plot(sv.anis)
```



The variogram map and directional semivariograms don't (in my mind) provide a definitive answer. The variogram map looks curious because going N (dY > 0) is quite different from going S (dY < 0). The directional semivariograms look reasonably similar in four directions, except perhaps that the sill is smaller in the 0 (N-S) direction. I will assume isotropy.

Fit some likely or possible VG models

```
sst.sph <- fit.variogram(sv.m, vgm(3, 'Sph', 400, 0.2))
sst.exp <- fit.variogram(sv.m, vgm(3, 'Exp', 400, 0.2))
sst.lin <- fit.variogram(sv.m, vgm(3, 'Lin', 400, 0.2))
sst.gau <- fit.variogram(sv.m, vgm(3, 'Gau', 400, 0.2))
sst.mat2 <- fit.variogram(sv.m, vgm(3, 'Mat', 50, 0.2, kappa=2))
sst.mat1 <- fit.variogram(sv.m, vgm(3, 'Mat', 100, 0.2, kappa=1))
sst.math <- fit.variogram(sv.m, vgm(3, 'Mat', 400, 0.2, kappa=0.5))
```

Choosing a variogram model

```
# Report variogram fit SSEs
c(Sph=attr(sst.sph, 'SSErr'),
  Exp=attr(sst.exp, 'SSErr'),
  Lin=attr(sst.lin, 'SSErr'),
  Mat2=attr(sst.mat2, 'SSErr'),
  Mat1=attr(sst.mat1, 'SSErr'),
  Mat0.5=attr(sst.math, 'SSErr') )
```

```
##          Sph          Exp          Lin         Mat2         Mat1
## 0.0078273578 0.0138982565 0.0064949241 0.0009145215 0.0020655417
##       Mat0.5
## 0.0138982565
```

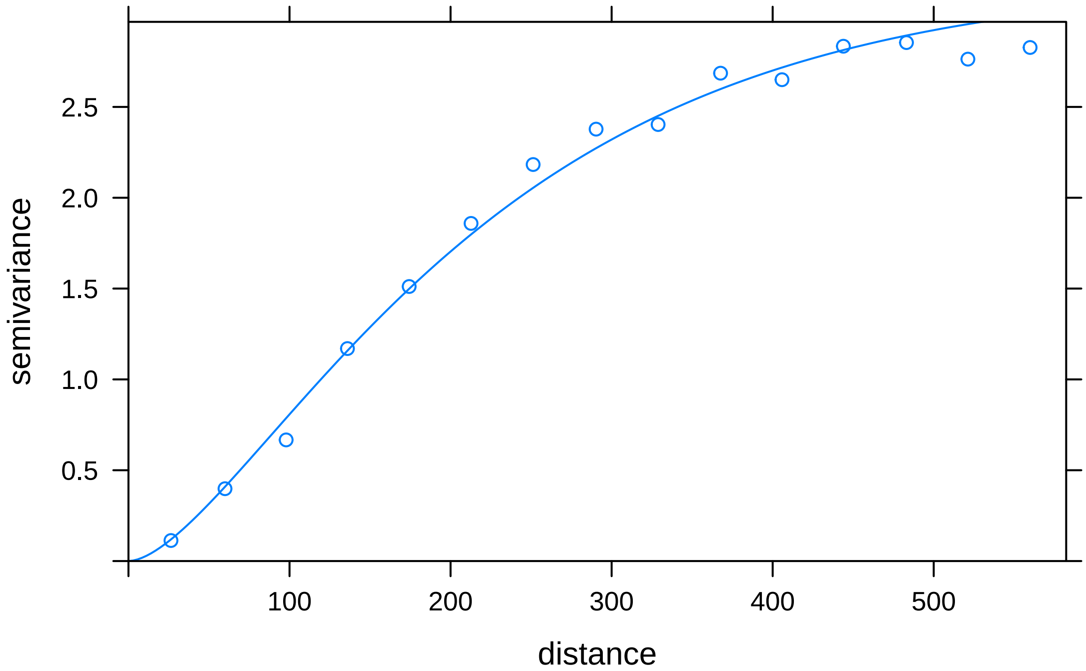
```
# report MS Pred Error for each model
sst.sphcv <- krige.cv(sst~long+lat, sst.sp, sst.sph)
sst.expcv <- krige.cv(sst~long+lat, sst.sp, sst.exp)
sst.lincv <- krige.cv(sst~long+lat, sst.sp, sst.lin)
sst.mat2cv <- krige.cv(sst~long+lat, sst.sp, sst.mat2)
sst.mat1cv <- krige.cv(sst~long+lat, sst.sp, sst.mat1)
sst.mathcv <- krige.cv(sst~long+lat, sst.sp, sst.math)
c(Sph=mean(sst.sphcv$residual^2),
  Exp=mean(sst.expcv$residual^2),
  Lin=mean(sst.lincv$residual^2),
  Mat2=mean(sst.mat2cv$residual^2),
  Mat1=mean(sst.mat1cv$residual^2),
  Mat0.5=mean(sst.mathcv$residual^2) )
```

```
##       Sph       Exp       Lin      Mat2      Mat1    Mat0.5
## 0.1514937 0.1548587        NA 0.1543949 0.1548170 0.1548587
```

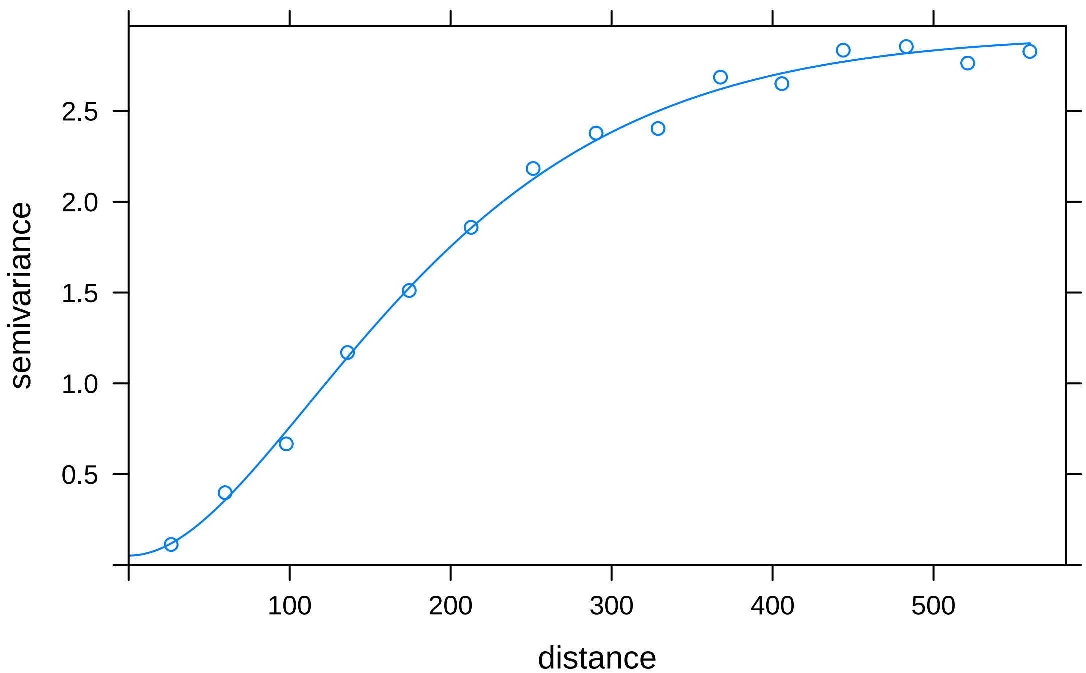
Matern, K=1 seems to fit semivariogram best, especially at short - intermediate distances. Very little difference in leave-one-out predicted values.

Overlay empirical semivariogram and fitted Matern model

```
plot(sv.m, sst.mat1)
```



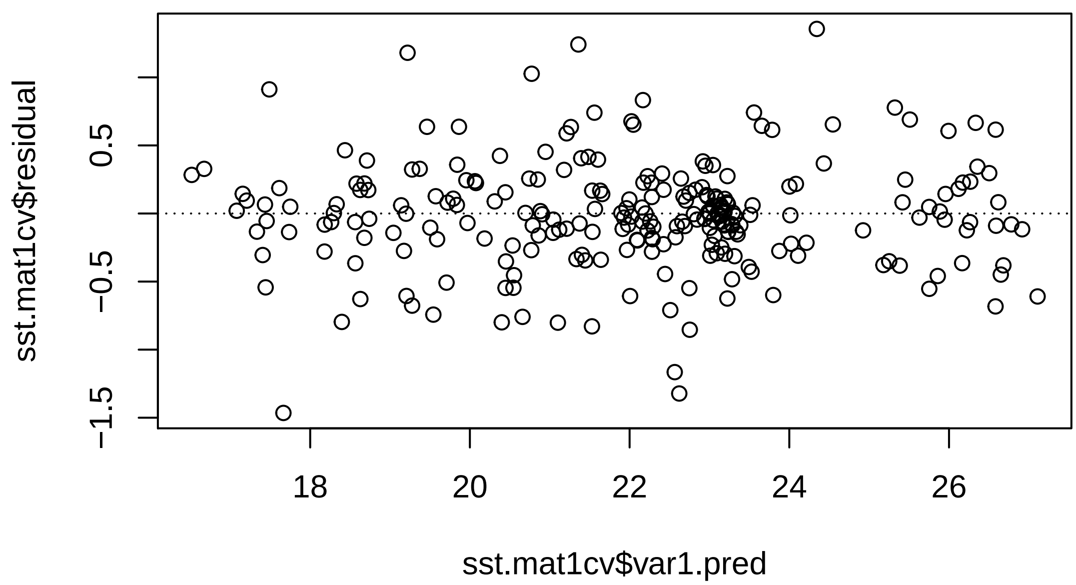

```r
plot(sv.m, sst.mat2)
```



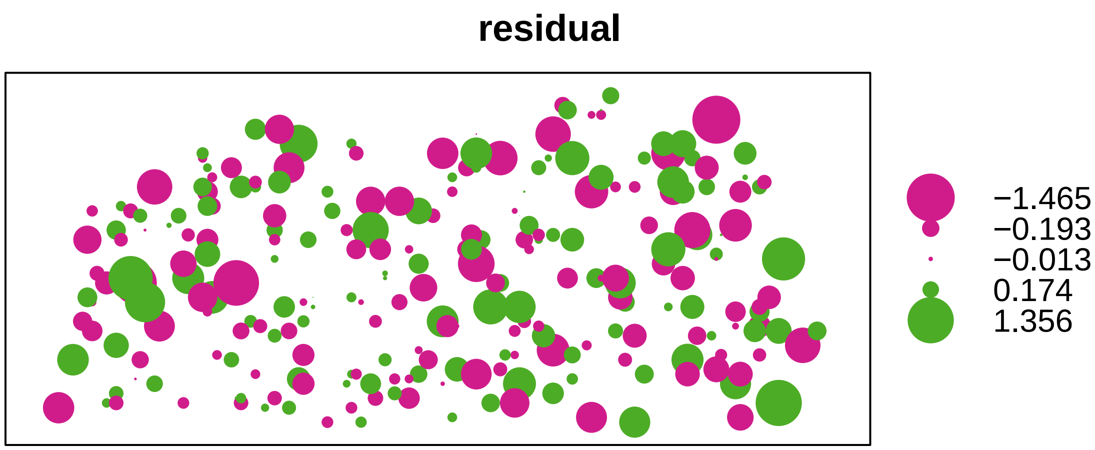
VG model looks good, especially at short and intermediate distances. Seems to estimate the sill, but that's less important for predictions (where want a good fit for small and intermediate distances).

Check residual plots

```
# residual vs predicted value plot
plot(sst.mat1cv$var1.pred, sst.mat1cv$residual)
abline(h=0, lty=3)
```




```
bubble(sst.mat1cv, 'residual')
```



Looks good - no sign of unequal variance (i.e. no trumpet shape) and no sign of lack of fit (smile or frown). Bubble plot has positive and negative residuals intermingled.

Make predictions for the four specific points

```
# create a data frame and spatial object with 4 specific locations
pts <- data.frame(long=c(-91.1, -86.2, -87.6, -85.1),
  lat=c(28.6, 26.35, 24.6, 24.0) )
pts.sp <- pts
coordinates(pts.sp) <- c('long', 'lat')
proj4string(pts.sp) <- CRS('+proj=longlat +datum=WGS84')

sst.pts <- krige(sst~long+lat, sst.sp, pts.sp, sst.mat2)
```

```
## [using universal kriging]
```

```
sst.pts$sd <- sqrt(sst.pts$var1.var)
sst.pts$var1.var <- NULL
sst.pts
```

```
##       coordinates var1.pred        sd
## 1  (-91.1, 28.6)  18.24757 0.4208755
## 2 (-86.2, 26.35)  25.96319 0.2754652
## 3  (-87.6, 24.6)  26.43370 0.3494618
## 4    (-85.1, 24)  26.61660 0.4809153
```

```
# cleaning up the output a bit
temp <- sst.pts
temp$sd <- NULL
temp$PredSST <- round(sst.pts$var1.pred, 2)
temp$var1.pred <- NULL
```

```r
temp$sd <- round(sst.pts$sd, 2)
print(temp)
```

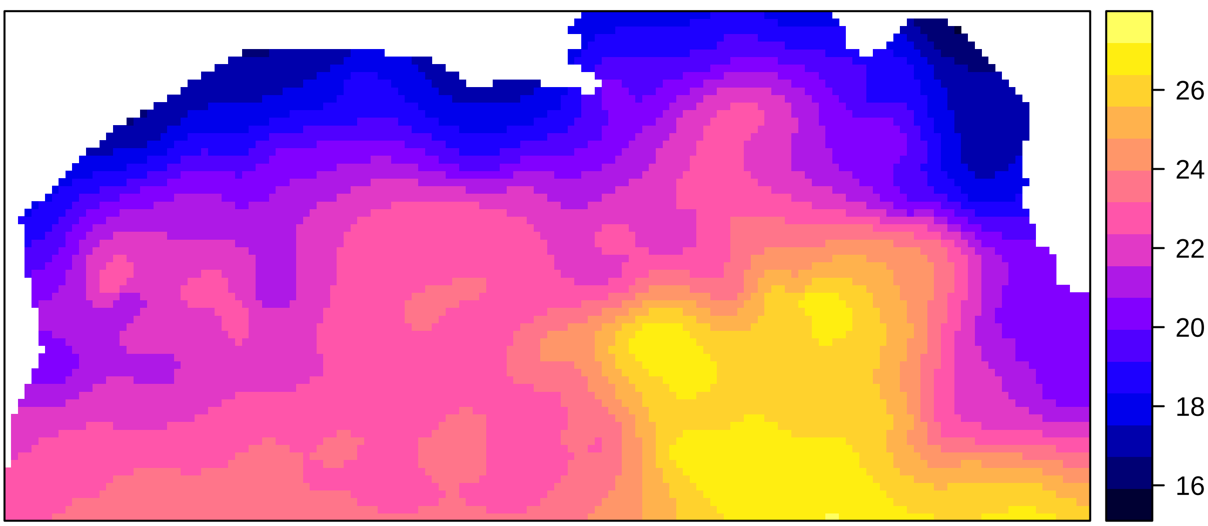
```
##      coordinates PredSST   sd
## 1  (-91.1, 28.6)   18.25 0.42
## 2 (-86.2, 26.35)   25.96 0.28
## 3  (-87.6, 24.6)   26.43 0.35
## 4    (-85.1, 24)   26.62 0.48
```

plot predictions on the fine grid

```r
sst.gridpred <- krige(sst~long+lat, sst.sp, finegrid.sp, sst.mat1)
```

```
## [using universal kriging]
```

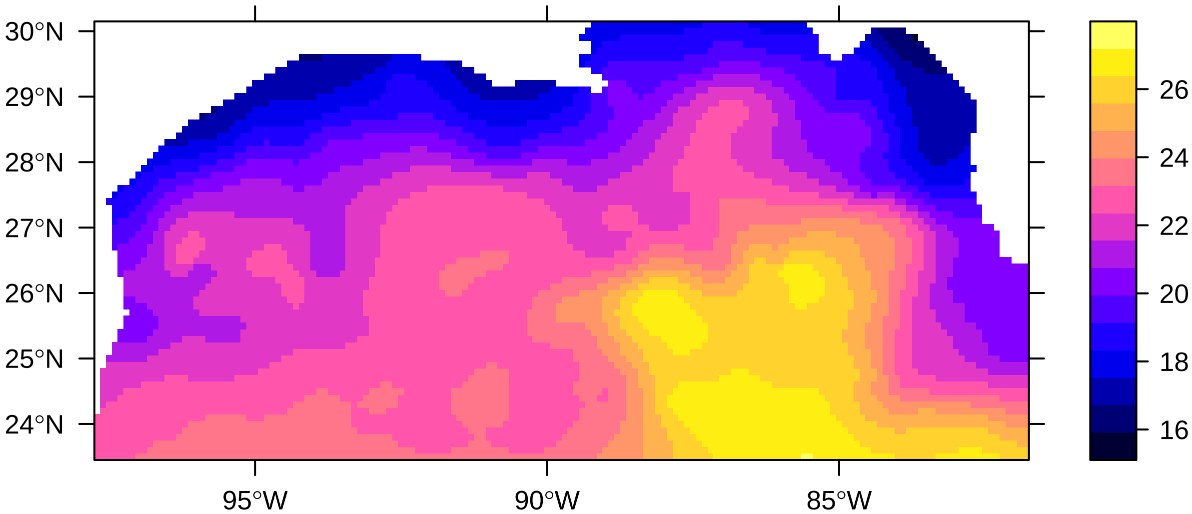
```r
finegrid.sp$predsst <- sst.gridpred$var1.pred
spplot(finegrid.sp, 'predsst')
```



The default color scheme looks reasonable here (actually showing cold to hot).

An extra credit point if you added axes to the plot

```r
spplot(finegrid.sp, 'predsst', scales=list(draw=T))
```


30°N
29°N
28°N
27°N
26°N
25°N
24°N
95°W
90°W
85°W
26
24
22
20
18
16