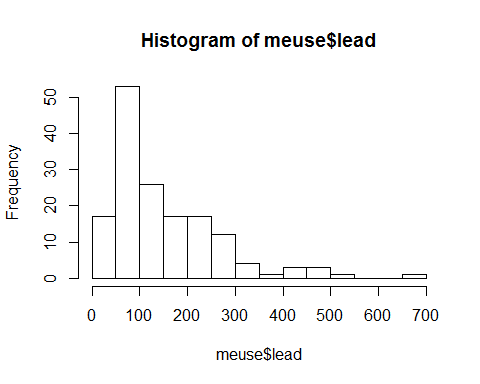
Lab 6

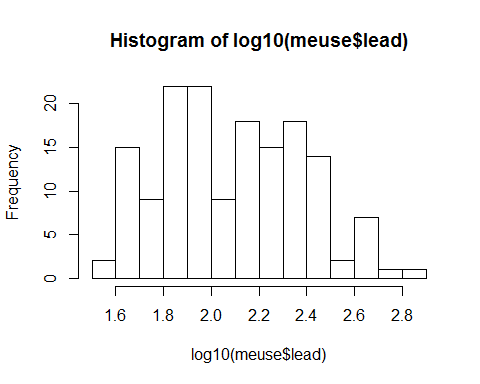
Dominic LaRoche

Tuesday, October 14, 2014

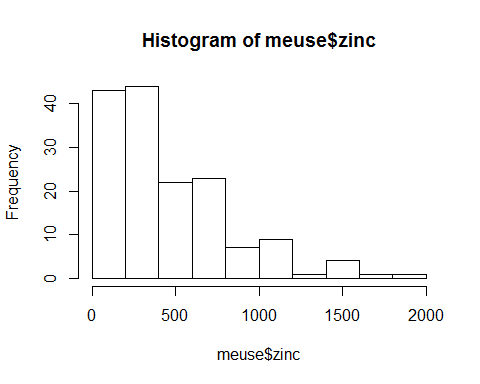
rm(list=ls())  
library(sp)  
library(gstat) #load the library  
library(lattice)  
library(pander)  
data(meuse.grid)  
data(meuse)  
#str(meuse.grid)  
  
hist(meuse$lead, breaks=12)



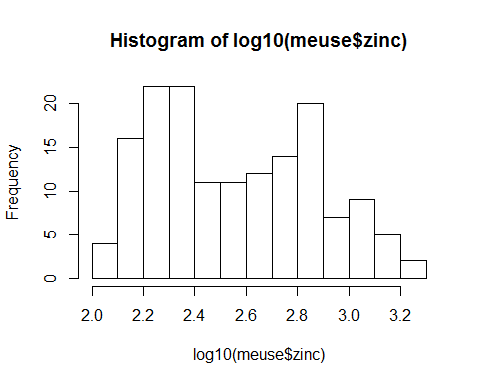
hist(log10(meuse$lead), breaks=12)



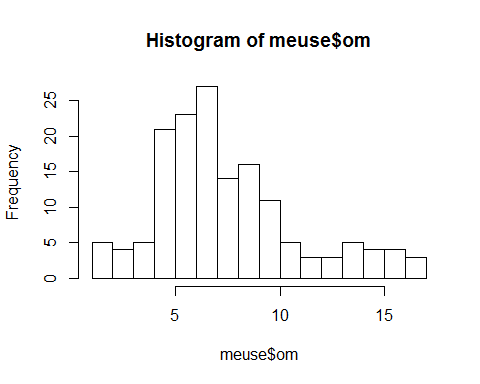
hist(meuse$zinc, breaks=12)



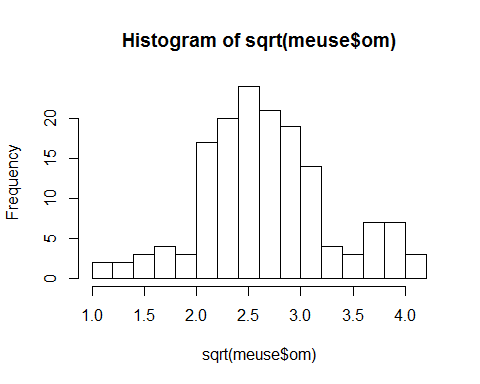
hist(log10(meuse$zinc), breaks=12)



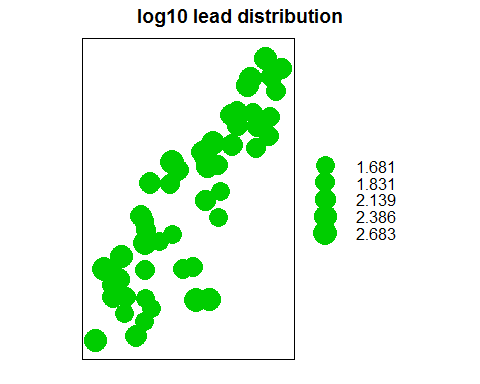
hist(meuse$om, breaks=12)



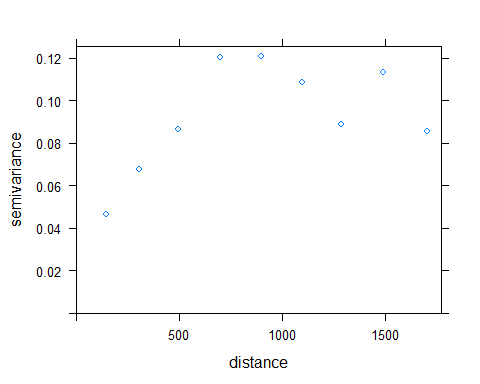
hist(sqrt(meuse$om), breaks=12)



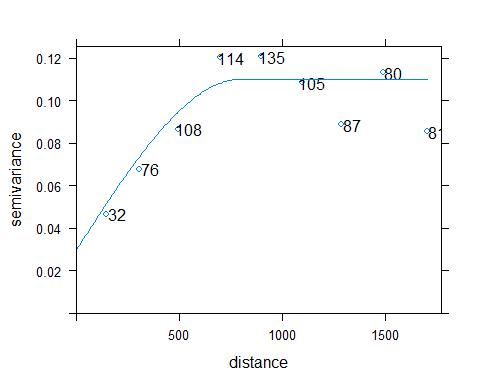
#ln tranformation of om won't really help here but sqrt seems to help stabilize the variance  
#Extract a subset of the target variable  
meuse.pb<-meuse[seq(1, length(meuse$lead), by=3), c('x', 'y', 'lead', 'zinc', 'om')]  
#str(meuse.pb)  
# rownames(meuse.pb)   
   
meuse.pb<-cbind(meuse.pb, ltpb=log10(meuse.pb$lead), ltzn=log10(meuse.pb$zinc))  
#Create a data frame of the lead observations at the extra points that have not been included in the subsample  
meuse.extra <- meuse[setdiff(rownames(meuse), rownames(meuse.pb)), c("x", "y", "lead")]  
meuse.extra <- cbind(meuse.extra, ltpb = log10(meuse.extra$lead))  
coordinates(meuse)<- ~x+y #convert data frame to "SpatialPointsDataFrame"  
coordinates(meuse.pb) <- ~ x + y  
coordinates(meuse.extra) <- ~ x + y  
coordinates(meuse.grid) <- ~ x + y  
# class(meuse)  
# summary(meuse.pb)  
bubble (meuse.pb, zcol= 'ltpb', main= 'log10 lead distribution')



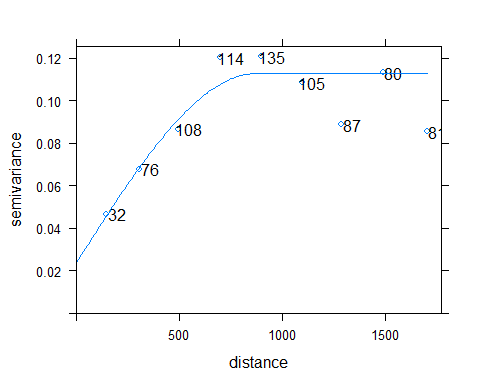
#variogram construction  
v.ltpb<-variogram(ltpb~1, data=meuse.pb, cutoff=1800, width=200)  
plot(v.ltpb)



m.ltpb<-vgm(psill=0.08, model='Sph', range =800, nugget=0.03)  
plot(v.ltpb, pl=T, model=m.ltpb)



m.ltpb.f <- fit.variogram(v.ltpb, m.ltpb)  
plot(v.ltpb, pl=T, model=m.ltpb.f)



#conduct ordinary kriging  
k.o <- krige(ltpb ~1, locations=meuse.pb, newdata=meuse.grid, model=m.ltpb.f)

## [using ordinary kriging]

# summary statistics  
summary(k.o)

## Object of class SpatialPointsDataFrame  
## Coordinates:  
## min max  
## x 178460 181540  
## y 329620 333740  
## Is projected: NA   
## proj4string : [NA]  
## Number of points: 3103  
## Data attributes:  
## var1.pred var1.var   
## Min. :1.71 Min. :0.0357   
## 1st Qu.:1.90 1st Qu.:0.0520   
## Median :2.03 Median :0.0608   
## Mean :2.06 Mean :0.0638   
## 3rd Qu.:2.23 3rd Qu.:0.0738   
## Max. :2.53 Max. :0.1119

#validation: prediction at the extra points  
k <- krige(ltpb ~ 1, meuse.pb, meuse.extra, m.ltpb.f)

## [using ordinary kriging]

# compute and summarize validation errors  
summary(k)

## Object of class SpatialPointsDataFrame  
## Coordinates:  
## min max  
## x 178605 181390  
## y 329714 333558  
## Is projected: NA   
## proj4string : [NA]  
## Number of points: 103  
## Data attributes:  
## var1.pred var1.var   
## Min. :1.74 Min. :0.0402   
## 1st Qu.:1.92 1st Qu.:0.0525   
## Median :2.11 Median :0.0580   
## Mean :2.10 Mean :0.0606   
## 3rd Qu.:2.26 3rd Qu.:0.0665   
## Max. :2.46 Max. :0.1032

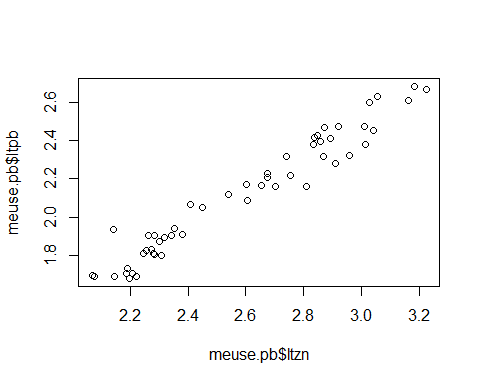
diff <- k$var1.pred - meuse.extra$ltpb  
summary(diff)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.6630 -0.0647 0.0215 0.0250 0.1270 0.3720

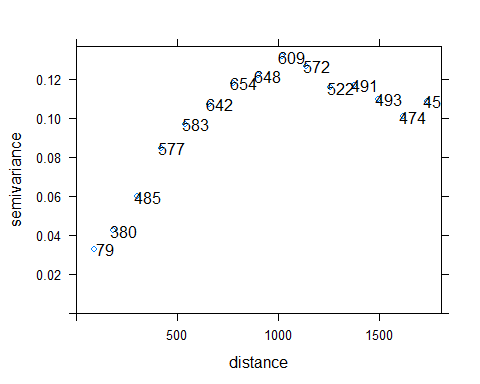
MSE\_ok<-sqrt(sum(diff^2)/length(diff))  
Bias\_ok<-sum(diff)/length(diff) # mean error (bias)  
med\_ok<-median(diff) # median error

cv\_ok<-krige.cv(ltpb ~1, locations=meuse.pb, model=m.ltpb.f, nfold=52)  
MSE\_okcv<-sqrt(sum(cv\_ok$residual^2)/length(cv\_ok$residual))  
Bias\_okcv<-sum(cv\_ok$residual)/length(cv\_ok$residual)  
med\_okcv<-median(cv\_ok$residual)  
#Created my own Loop for n-fold cross validation of ordinary kriging  
#I wanted to see whether this would produce similar results   
data(meuse)  
meuse.pbN<-meuse[seq(1, length(meuse$lead), by=3), c('x', 'y', 'lead', 'zinc')]  
NfoldDiff<-vector()  
for(i in 1:length(meuse.pb$lead)){  
 #remove one case of the target variable  
 meuse.pbn<-meuse.pbN[-i, c('x', 'y', 'lead', 'zinc')]  
 meuse.pbn<-cbind(meuse.pbn, ltpb=log10(meuse.pbn$lead), ltzn=log10(meuse.pbn$zinc))  
 meuse.pb.extra <- meuse.pbN[i, c("x", "y", "lead")]  
 meuse.pb.extra <- cbind(meuse.pb.extra, ltpb = log10(meuse.pb.extra$lead))  
 coordinates(meuse.pbn) <- ~ x + y  
 coordinates(meuse.pb.extra) <- ~ x + y  
 #variogram construction on n-1 data points  
 v.ltpb<-variogram(ltpb~1, data=meuse.pbn, cutoff=1800, width=200)  
 m.ltpb<-vgm(psill=0.08, model='Sph', range =800, nugget=0.03)  
 m.ltpb.f <- fit.variogram(v.ltpb, m.ltpb)  
 #conduct ordinary kriging  
 k <- krige(ltpb ~ 1, meuse.pbn, meuse.pb.extra, m.ltpb.f)  
 NfoldDiff[i] <- meuse.pb.extra$ltpb - k$var1.pred   
}  
summary(NfoldDiff)  
MSEmyCVdiff<-sqrt(sum(NfoldDiff^2)/length(NfoldDiff))  
BiasmyCVdiff<-sum(NfoldDiff)/length(NfoldDiff) # mean error (bias)  
medmyCVdiff<-median(NfoldDiff)

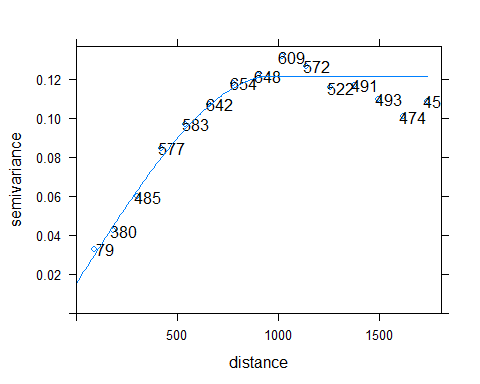
#correlation  
cor(meuse.pb$ltzn, meuse.pb$ltpb)  
plot(meuse.pb$ltzn, meuse.pb$ltpb)



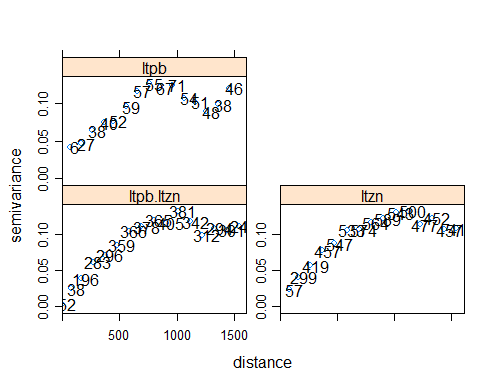
#extract the co-variate  
meuse.co<-as.data.frame(meuse)[, c('x', 'y', 'zinc')]  
meuse.co <- cbind(meuse.co, ltzn = log10(meuse.co$zinc))  
coordinates(meuse.co) <- ~ x + y  
#variogram  
v.ltzn <- variogram(ltzn ~ 1, meuse.co, cutoff=1800)  
plot(v.ltzn, pl=T)



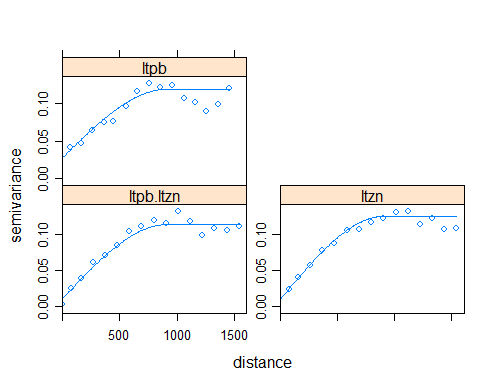
m.ltzn <- vgm(psill=.11, model="Sph", range=1000, nugget=.02) #by visual inspection  
m.ltzn.f <- fit.variogram(v.ltzn, m.ltzn) #by least square fit  
plot(v.ltzn, pl=T, model=m.ltzn.f)



#Model the coregionalization: create a gstat structure containing sets of data: subsample for lead   
#and full sample for zinc  
g1 <- gstat(NULL, id = "ltpb", form = ltpb ~ 1, data = meuse.pb)  
g1 <- gstat(g1, id = "ltzn", form = ltzn ~ 1, data = meuse.co)  
#variogram and cross-variogram  
v.cross <- variogram(g1)  
plot(v.cross, pl=T)

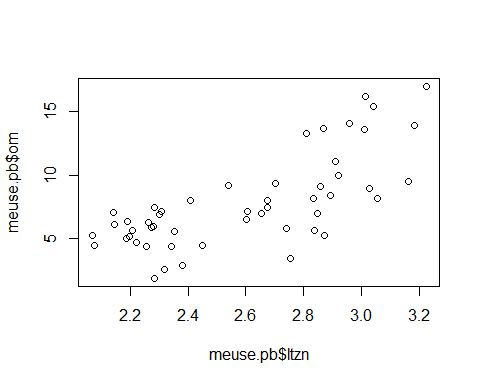


#Fit cross-variogram model  
g1 <- gstat(g1, id = "ltpb", model = m.ltpb.f, fill.all=T)  
g1 <- fit.lmc(v.cross[-16,], g1)  
plot(variogram(g1), model=g1$model)

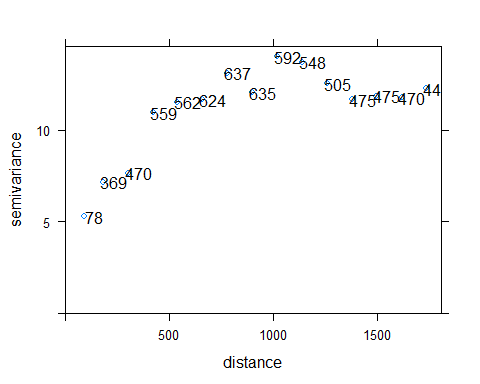


#Prediction at the grids  
k.c1 <- predict.gstat(g1, meuse.grid)  
summary(k.c1$ltpb.pred); summary(k.c1$ltpb.var)  
#Add to DF for plotting  
meuse.grid$k.cl<-k.c1$ltpb.pred  
meuse.grid$k.cl.se<-sqrt(k.c1$ltpb.var)  
##Validation  
#Interpolate at extra points  
k <- predict.gstat(g1, meuse.extra)  
codiff <- k$ltpb.pred - meuse.extra$ltpb  
summary(codiff)  
MSEcodiff<-sqrt(sum(codiff^2)/length(codiff)) # RMS error (precision)  
Biascodiff<-sum(codiff)/length(codiff) # mean error (bias)  
medcodiff<-median(codiff) # median error  
  
#Cross Validation  
cv.c2 <- gstat.cv(g1, nfold=52)  
pe2<-cv.c2$residual  
# summary(cv.c2$residual)  
MSEcvcodiff<-sqrt(mean(cv.c2$residual^2))  
# mean(cv.c2$residual)  
Biascvcodiff<-sum(cv.c2$residual)/length(cv.c2$residual)  
medcvcodiff<-median(cv.c2$residual)

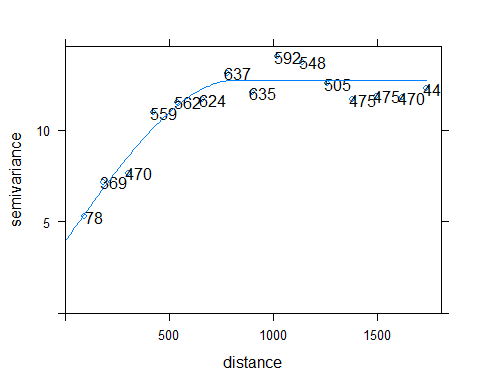
#correlation  
cor(meuse.pb$ltzn, meuse.pb$om,use="complete.obs")  
plot(meuse.pb$ltzn, meuse.pb$om)



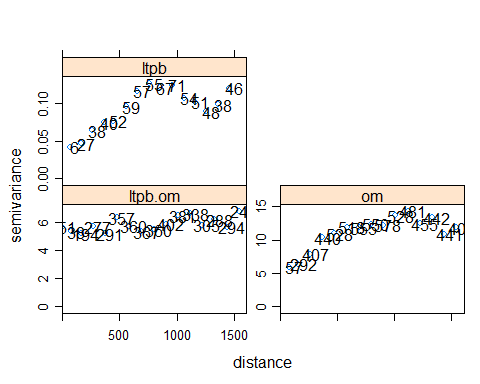
#extract the co-variate  
meuse.co<-na.omit(as.data.frame(meuse)[, c('x', 'y', 'om')])  
coordinates(meuse.co) <- ~ x + y  
#variogram  
v.om <- variogram(om ~ 1, meuse.co, cutoff=1800)  
plot(v.om, pl=T)



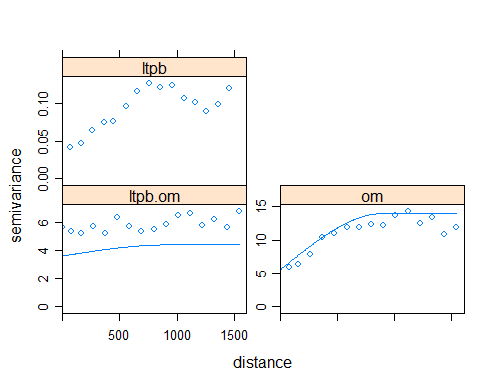
m.om <- vgm(psill=.12, model="Sph", range=800, nugget=.02) #by visual inspection  
m.om.f <- fit.variogram(v.om, m.om) #by least square fit  
plot(v.om, pl=T, model=m.om.f)



#Model the coregionalization: create a gstat structure containing sets of data: subsample for lead   
#and full sample for zinc  
g1 <- gstat(NULL, id = "ltpb", form = ltpb ~ 1, data = meuse.pb)  
g1 <- gstat(g1, id = "om", form = om ~ 1, data = meuse.co)  
#variogram and cross-variogram  
v.cross <- variogram(g1)  
plot(v.cross, pl=T)



#Fit cross-variogram model  
g1 <- gstat(g1, id = "ltpb", model = m.ltpb.f, fill.all=T)  
g1 <- fit.lmc(v.cross[-16,], g1)  
plot(variogram(g1), model=g1$model)



#Prediction at the grids  
k.c2 <- predict.gstat(g1, meuse.grid)  
summary(k.c2$ltpb.pred); summary(k.c2$ltpb.var)  
#Add to DF for plotting  
meuse.grid$k.cl<-k.c1$ltpb.pred  
meuse.grid$k.cl.se<-sqrt(k.c1$ltpb.var)  
##Validation  
#Interpolate at extra points  
k2 <- predict.gstat(g1, meuse.extra)  
codiff <- k2$ltpb.pred - meuse.extra$ltpb  
summary(codiff)  
MSEco2<-sqrt(sum(codiff^2)/length(codiff)) # RMS error (precision)  
Biasco2<-sum(codiff)/length(codiff) # mean error (bias)  
medco2<-median(codiff) # median error  
  
#Cross Validation  
cv.c2 <- gstat.cv(g1, nfold=52)  
pe2.2<-cv.c2$residual  
# summary(cv.c2$residual)  
MSEcvco2<-sqrt(mean(cv.c2$residual^2))  
# mean(cv.c2$residual)  
Biascvco2<-sum(cv.c2$residual)/length(cv.c2$residual)  
medcvco2<-median(cv.c2$residual)

# Assignment 1, 2 and 3

For the ease of interpretation and comparison root-mean-squared errors and biases for all methods are presented in Table 1. The estimate of prediction accuracy using the extra points is lower than the estimate usinf n-fold cross-validation. I used the built in cross-validation method and manually coded cross-validation for the ordinary kriging prediction. Results from these two methods were very similar but not identical. The RMSE from n-fold cross validation is likely to be a better estimate of prediction error on a new a sample since it more systematically addresses the uncertainty.

The use of co-kriging with zinc greatly reduced the prediction error when compared to ordinary kriging. The predictions and standard errors for ordinary kriging and co-kriging are given in figures 1 and 2. These figures show the better prediction accuracy of co-kriging with zinc. However, co-kriging with 'om' rather than zinc reduced the prediction accuracy (Table 1). This is because the correlation between 'om' and lead was not nearly as strong as the correlation between zinc and lead (0.72 vs 0.97).

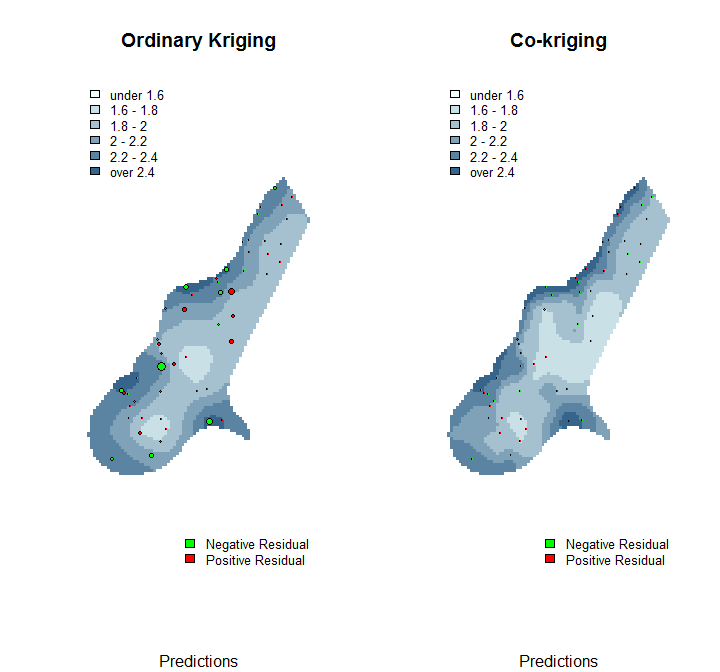
library(pander)  
sumtab<-matrix(c(MSE\_ok,MSE\_okcv,MSEmyCVdiff,MSEcvcodiff,MSEcvco2,Bias\_ok,Bias\_okcv,BiasmyCVdiff,Biascvcodiff,Biascvco2),5,2)  
colnames(sumtab)<-c("RMSE","Bias")  
rownames(sumtab)<-c("Ordinary Kriging","Ordinary Kriging (CV)","Ord. Krig. (manual CV)","Co-kriging with Zinc","Co-kriging with 'om'")  
pandoc.table(sumtab,caption="Root-mean-squared errors and biases for all methods.")

|  |  |  |
| --- | --- | --- |
|  | RMSE | Bias |
| **Ordinary Kriging** | 0.166 | 0.02502 |
| **Ordinary Kriging (CV)** | 0.2369 | 0.00367 |
| **Ord. Krig. (manual CV)** | 0.2414 | 0.002854 |
| **Co-kriging with Zinc** | 0.0711 | -2.633e-06 |
| **Co-kriging with 'om'** | 1.17 | -0.005124 |

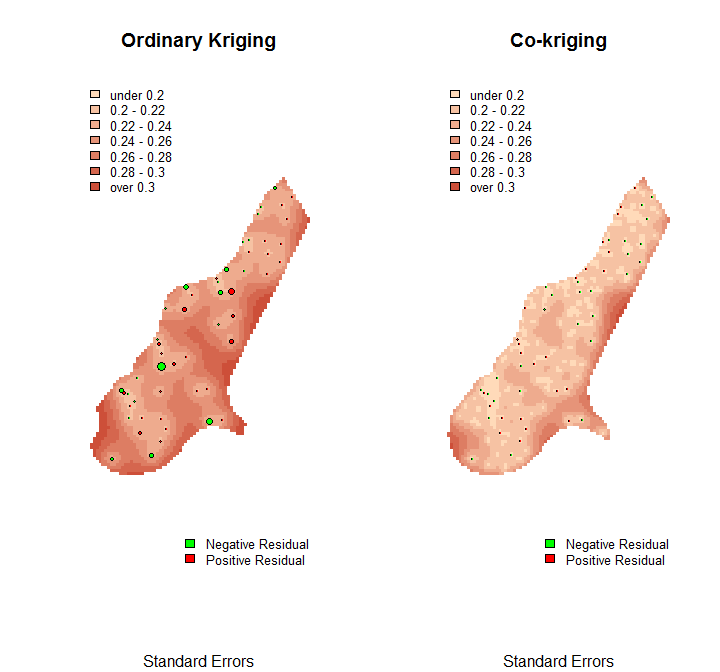
Root-mean-squared errors and biases for all methods.

## Checking rgeos availability: TRUE

meuse.grid.df<-SpatialPixelsDataFrame(points=meuse.grid@coords,data=as.data.frame(meuse.grid))  
  
#predict over entire area  
OK\_fit<-gstat(id='OK\_fit', formula=ltpb ~ 1, data=meuse.pb, model=m.ltpb.f)   
pe<-gstat.cv(OK\_fit, nfold=52, debug.level=0, random=FALSE)$residual   
ok<-predict(OK\_fit, newdata=meuse.grid.df, debug.level=0)   
meuse.grid.df$ok\_pred<-ok$OK\_fit.pred  
meuse.grid.df$ok\_se<-sqrt(ok$OK\_fit.var)  
  
par(mfrow=c(1,2))  
image(meuse.grid.df, 'ok\_pred', breaks=brks, col=cols)  
symbols(meuse.pbN[c("x","y")], circles=abs(pe)\*100, fg='black', bg=scols[(pe<0)+1], inches=FALSE, add=TRUE)  
legend('topleft',fill=cols, legend=leglabs(brks), bty ='n', cex=0.8)  
legend('bottomright', fill=scols, legend=c("Negative Residual","Positive Residual"), bty='n', cex=0.8)  
title("Ordinary Kriging","Predictions")  
  
image(meuse.grid.df, 'k.cl', breaks=brks, col=cols)  
symbols(meuse.pbN[c("x","y")], circles=abs(pe2)\*100, fg='black', bg=scols[(pe2<0)+1], inches=FALSE, add=TRUE)  
legend('topleft',fill=cols, legend=leglabs(brks), bty ='n', cex=0.8)  
legend('bottomright', fill=scols, legend=c("Negative Residual","Positive Residual"), bty='n', cex=0.8)  
title("Co-kriging","Predictions")



#plot SE's  
image(meuse.grid.df, 'ok\_se', breaks=brks.se, col=cols.se)  
symbols(meuse.pbN[c("x","y")], circles=abs(pe)\*100, fg='black', bg=scols[(pe<0)+1], inches=FALSE, add=TRUE)  
legend('topleft',fill=cols.se, legend=leglabs(brks.se), bty ='n', cex=0.8)  
legend('bottomright', fill=scols, legend=c("Negative Residual","Positive Residual"), bty='n', cex=0.8)  
title("Ordinary Kriging","Standard Errors")  
  
image(meuse.grid.df, 'k.cl.se', breaks=brks.se, col=cols.se)  
symbols(meuse.pbN[c("x","y")], circles=abs(pe2)\*100, fg='black', bg=scols[(pe2<0)+1], inches=FALSE, add=TRUE)  
legend('topleft',fill=cols.se, legend=leglabs(brks.se), bty ='n', cex=0.8)  
legend('bottomright', fill=scols, legend=c("Negative Residual","Positive Residual"), bty='n', cex=0.8)  
title("Co-kriging","Standard Errors")



par(mfrow=c(1,1))