Topic 2: Model diagnostic

This topic applies the previously considered methods of spatial extrapolation to model diagnostics. In particular, we consider

- Introduction to **model diagnostic**.
- **Leave-one-out cross validation.**
- ***N*-fold cross validation.**

Model Diagnostics

The model diagnostics we have seen so far are fitted and residual plots, visual and numerical inspection of variogram models, and visual and numerical inspection of kriging results. Along the way, we have seen many model decisions that needed to be made; the major ones being the following:

1. Possible transformation of the dependent variable.
2. The form of the trend function.
3. Possibly directional dependence for the sample variogram.
4. The variogram model type.
5. The variogram model coefficient values, or fitting method.

There was fairly little statistical guidance to explain which choices are better. To some extent we can 'ask' the data what a good decision is, and for this we may use cross validation.

Cross validation splits the data set into two sets: a **modelling set** and a **validation set.** The modelling set is used for variogram modelling and kriging on the locations of the validation set, and then the validation measurements can be compared to their predictions. If all went well, cross validation residuals are small, have zero mean, and no apparent structure.

How should we choose or isolate a set for validation? Usually it done by randomly partitioning the data in a model and test set.

Let us try this for the MEUSE data set, splitting it in 100 observations for modelling and 55 for testing:

```
> library(gstat)
> data(meuse)
> coordinates(meuse) <- c("x", "y")
> sel100 <- sample(1:155, 100)
> m.model <- meuse[sel100,]
> m.valid <- meuse[-sel100,]
```

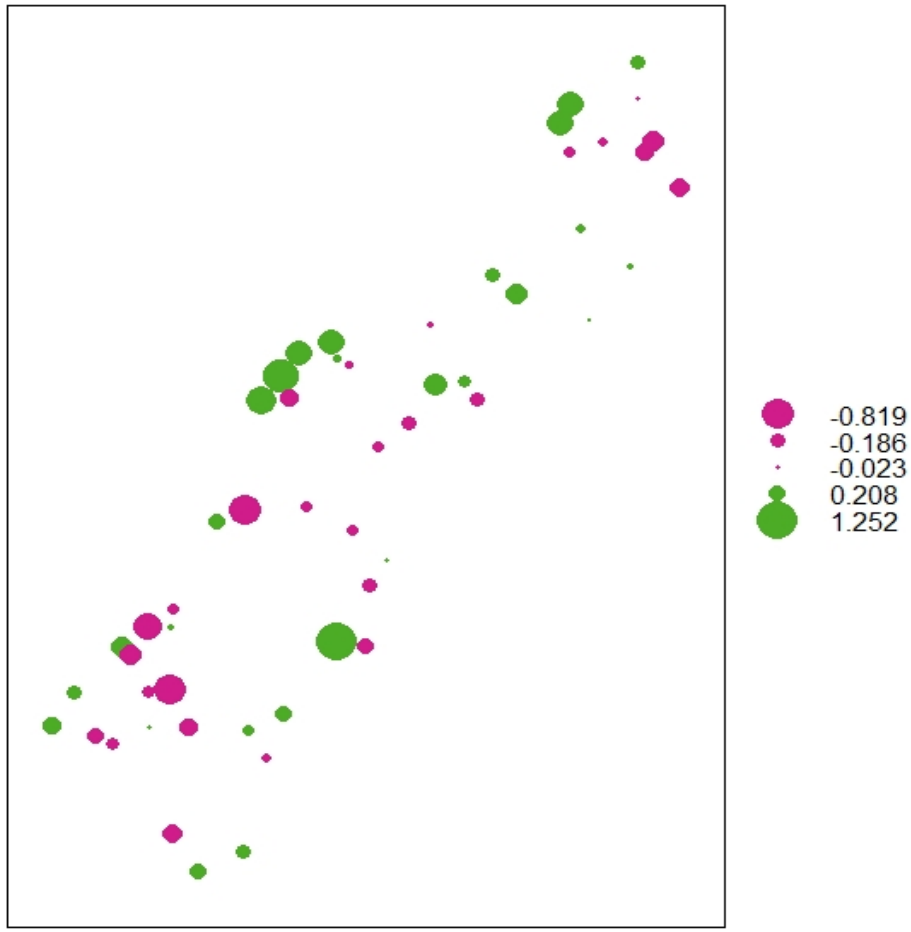
Then we fit a variogram model and compute the R^2 statistic of the residuals:

```
> v100.fit <- fit.variogram(variogram(log(zinc)~1, m.model),
+ vgm(1, "Sph", 800, 1))
> m.valid.pr <- krige(log(zinc)~1, m.model,m.valid,v100.fit)
> resid.kr <- log(m.valid$zinc) - m.valid.pr$var1.pred
> summary(resid.kr)
Min.   1st Qu.    Median      Mean   3rd Qu.  Max.
-0.768695 -0.176018  0.007515  0.025891  0.216754 0.913719
> resid.mean <- log(m.valid$zinc) - mean(log(m.valid$zinc))
> R2 <- 1 - sum(resid.kr^2)/sum(resid.mean^2)
> R2
[1] 0.7199791
```

which indicates that kriging prediction is a better predictor than the mean, as R^2 is 0.72. Running this analysis again can result in different values, as another random sample is chosen.

A map with cross validation residuals can be obtained by

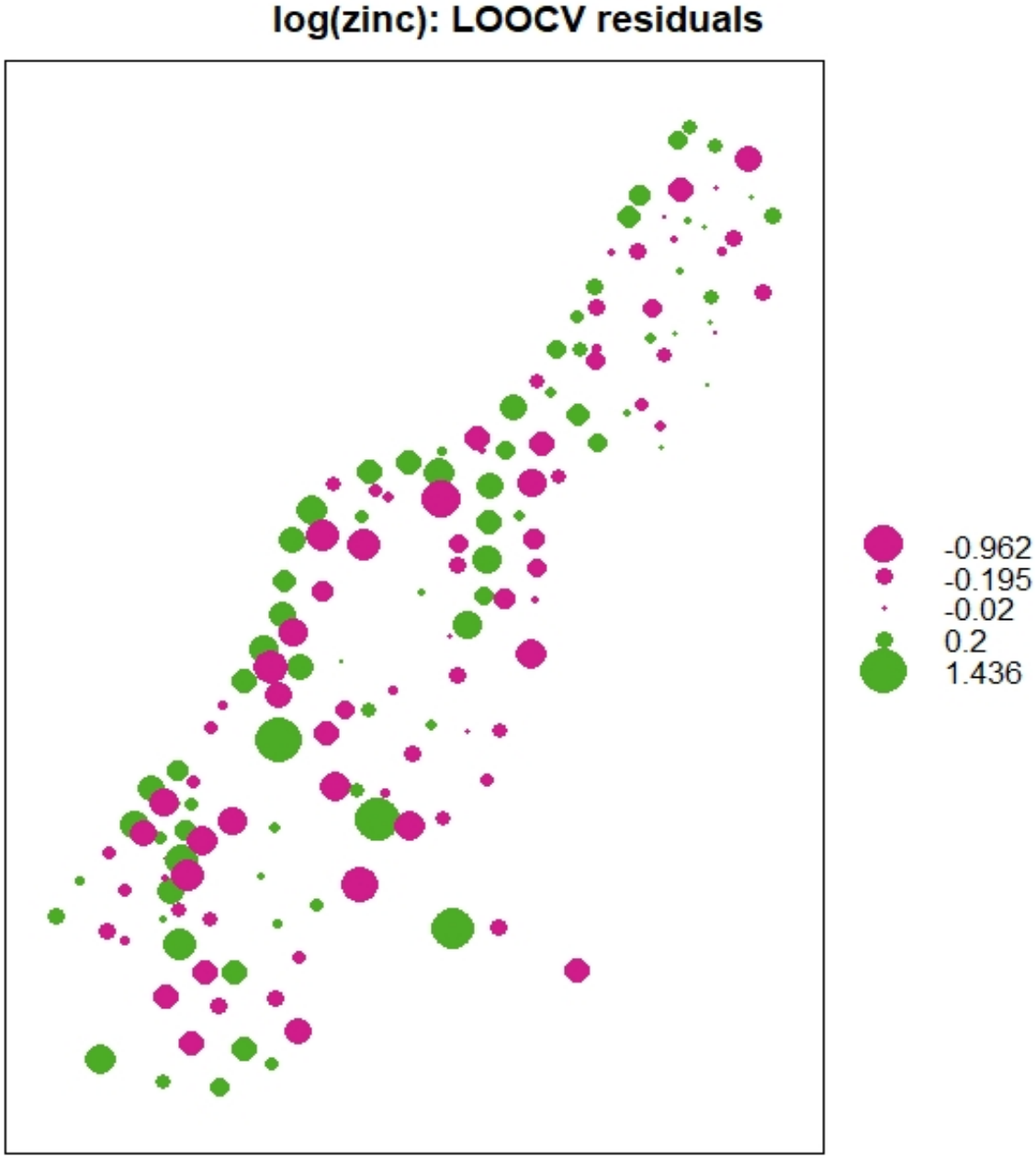
```
> m.valid.pr$res <- resid.kr
> bubble(m.valid.pr, "res")
```



A more automated way to do this is provided by the GSTAT function **krige.cv** for univariate cross validation:

- **Leave-one-out cross validation (LOOCV)** visits a data point, and predicts the value at that location by leaving out the observed value, and proceeds with the next data point.

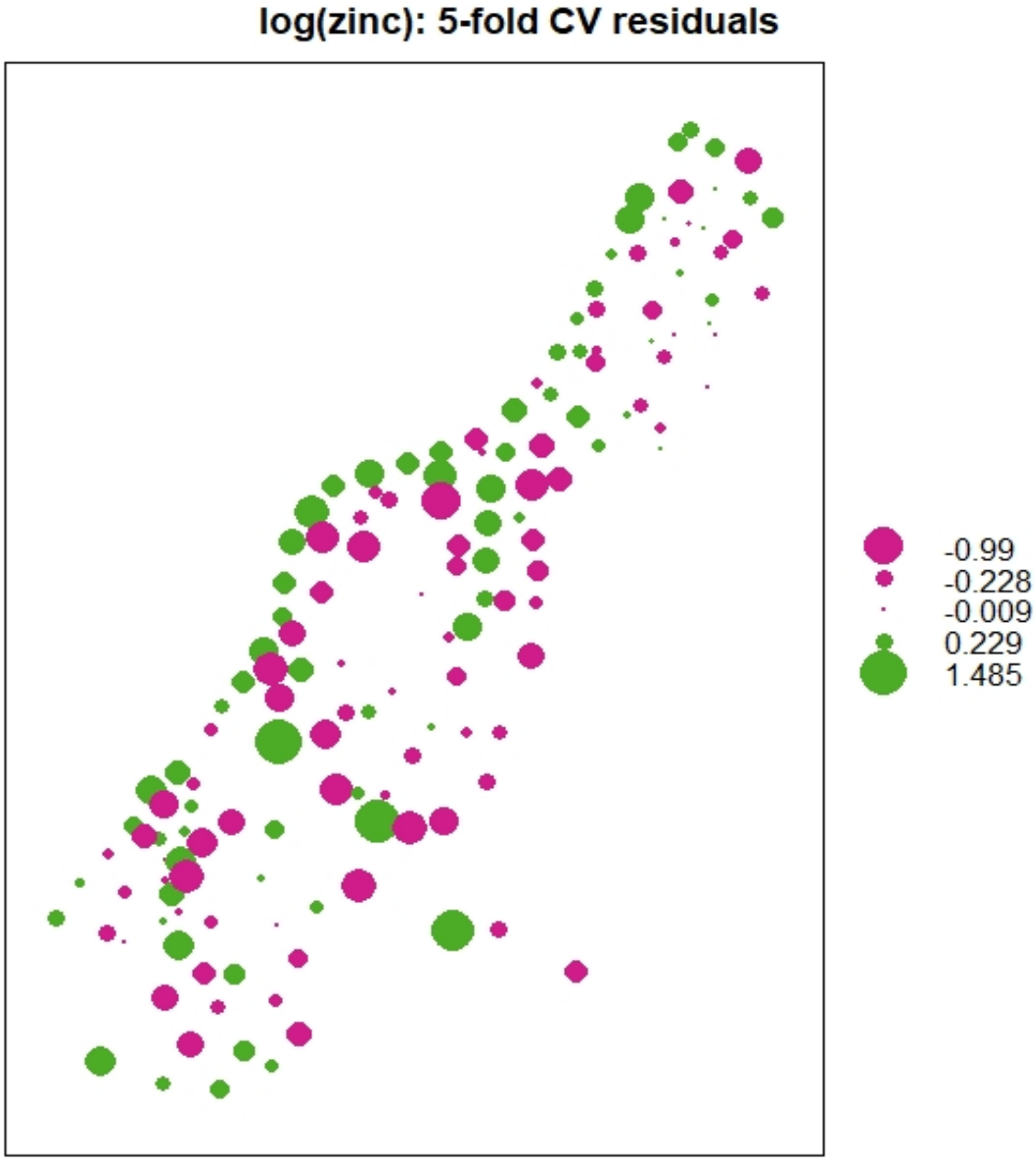
```
> v.fit <- vgm(.59, "Sph", 874, .04)
> cvLOOCV <- krige.cv(log(zinc)~1, meuse, v.fit,
+ nfold=nrow(meuse))
|======================================| 100%
> bubble(cvLOOCV, "residual", main = "log(zinc):
+ LOOCV residuals")
```


log(zinc): LOOCV residuals
-0.962
-0.195
-0.02
0.2
1.436

- ***N*-fold cross validation** makes a partition of a data set into N parts. For all observation in a part, predictions are made based on the remaining $N - 1$ parts. This is repeated for each of the N parts.

```
> cv155 <- krige.cv(log(zinc)~1, meuse, v.fit, nfold=5)
|=========================================| 100%
> bubble(cv155, "residual", main = "log(zinc):
+ 5-fold CV residuals")
```

For a large number of locations the N-fold cross validation may be much faster than LOOCV.


log(zinc): 5-fold CV residuals
-0.99
-0.228
-0.009
0.229
1.485

Key R commands

polygon(x, y)	*draws the polygons with vertices given in x and y*
krige.conv(geodata,...)	*performs spatial prediction by using a given covariance*
expand.grid(...)	*creates a data frame from all combinations of the supplied vectors*
image(x,...)	*images gridded data*
krige(formula,...)	*simple, ordinary or universal kriging*
sample(x, size)	*takes a sample of size from x*
dataframe[-set,]	*selects rows of dataframe except those in set*
krige.cv(formula,...)	*a cross validation function for kriging*
bubble(obj, ...)	*creates a bubble plot of spatial data*