Ordinary Kriging in QGIS and R

Renate Thiede – Spatial Data Science

Fundamentals: What is Kriging?

- A method of spatial prediction, more specifically spatial interpolation, which uses the distances and spatial correlation between known points to predict the value of a function at unknown points based on a weighted average.
- Kriging consists of two steps:
  1. Fitting a variogram to the points
  2. Using the variogram to calculate the weights and interpolate
- The result of Kriging is an interpolated surface. The value at each point is the Best Linear Unbiased Prediction (BLUP) at that point, i.e. the interpolated value at each point minimises the error at that point.

Fundamentals: What is Kriging?

In mathematical terms, the idea of Kriging is to predict the value of a function $Y(S_0)$ with a linear predictor, where S_0 is a location in geographic space.

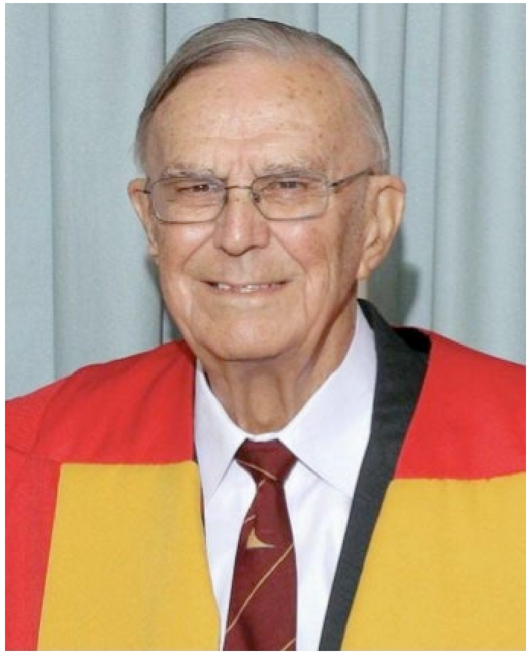
We would like unbiased predictor i.e. $E[T] = E[Y(S_0)]$. We also want the **best** predictor i.e. $var(T - Y(S_0))$ should be the lowest among all linear unbiased predictors. The Kriging predictor is the BLUP.

STEP 1: We estimate the mean $[\hat{\mu} = \left(X^T\, C^{-1}\, X\right)^{-1} X^T\, C^{-1}\, y]$, where X is a vector of n elements all equal to 1, C is an $n \times\ n$ covariance matrix, and y is a vector of the n data values.

STEP 2: Obtain the predictor $T\ = x_0^T \hat{\mu} + c_0^T\, C^{-1}(y - X\hat{\mu})$, where c_0 is the covariance function evaluated between $Y(S_0)$ and the variables at the observed locations $Y(u_i)$, and x_0 is the vector with one element equal to 1.

Fundamentals: History of Kriging

- Kriging is rooted in South Africa. It is named after Danie G Krige, a renowned South African engineer and statistician. He was a professor at the University of the Witwatersrand.
- His 1951 paper in the *Journal of the Chemical, Metallurgical and Mining Society of South Africa* first introduced the technique.
- The theory was developed further by Georges Matheron, who coined the term ‘Kriging’.



Prof Danie Krige, 1919 – 2013
Photo credit: https://www.saimm.co.za/saimm-events/upcoming-events/the-danie-krige-memorial-lecture

Fundamentals: Types of Kriging

There are many different types of Kriging, suitable for different kinds of data. While this course only introduces Ordinary Kriging, a list of common Kriging methods is presented below for completeness.

1. Ordinary Kriging
2. Universal Kriging
3. Simple Kriging
4. Indicator Kriging
5. Probability Kriging
6. Disjunctive Kriging
7. Empirical Bayesian Kriging
8. CoKriging

Fundamentals: the Ordinary Kriging model

The model for Ordinary Kriging is as follows:

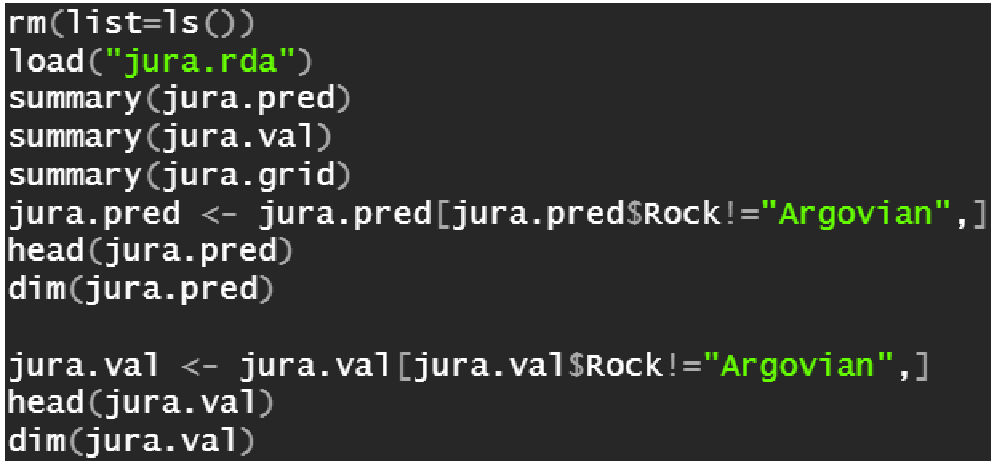
$$Y(u) = m + \epsilon(u)$$

where $Y(u)$ is the estimated value at location u, m is the mean, and $\epsilon(u)$ is the error term. This should look similar to linear regression, which after all is simply another prediction model.

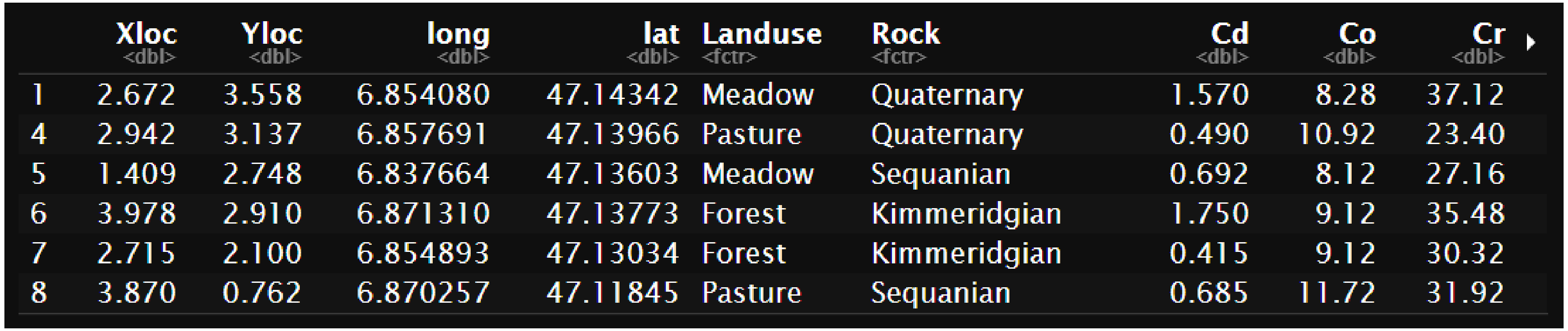
- The mean m is assumed constant but must be estimated.
- Ordinary Kriging focuses on prediction of deviations from the unknown mean.

Let’s do it in R: Importing the data

Import the `jura.rda` file. This gives the concentrations of various chemical elements at sampled points.



```r
rm(list=ls())
load("jura.rda")
summary(jura.pred)
summary(jura.val)
summary(jura.grid)
jura.pred <- jura.pred[jura.pred$Rock!="Argovian",]
head(jura.pred)
dim(jura.pred)

jura.val <- jura.val[jura.val$Rock!="Argovian",]
head(jura.val)
dim(jura.val)
```



```
    Xloc   Yloc   long       lat       Landuse  Rock         Cd     Co     Cr   ▸
   <dbl>  <dbl>  <dbl>      <dbl>     <fctr>   <fctr>       <dbl>  <dbl>  <dbl>
1  2.672  3.558  6.854080  47.14342  Meadow   Quaternary   1.570   8.28  37.12
4  2.942  3.137  6.857691  47.13966  Pasture  Quaternary   0.490  10.92  23.40
5  1.409  2.748  6.837664  47.13603  Meadow   Sequanian    0.692   8.12  27.16
6  3.978  2.910  6.871310  47.13773  Forest   Kimmeridgian 1.750   9.12  35.48
7  2.715  2.100  6.854893  47.13034  Forest   Kimmeridgian 0.415   9.12  30.32
8  3.870  0.762  6.870257  47.11845  Pasture  Sequanian    0.685  11.72  31.92
```

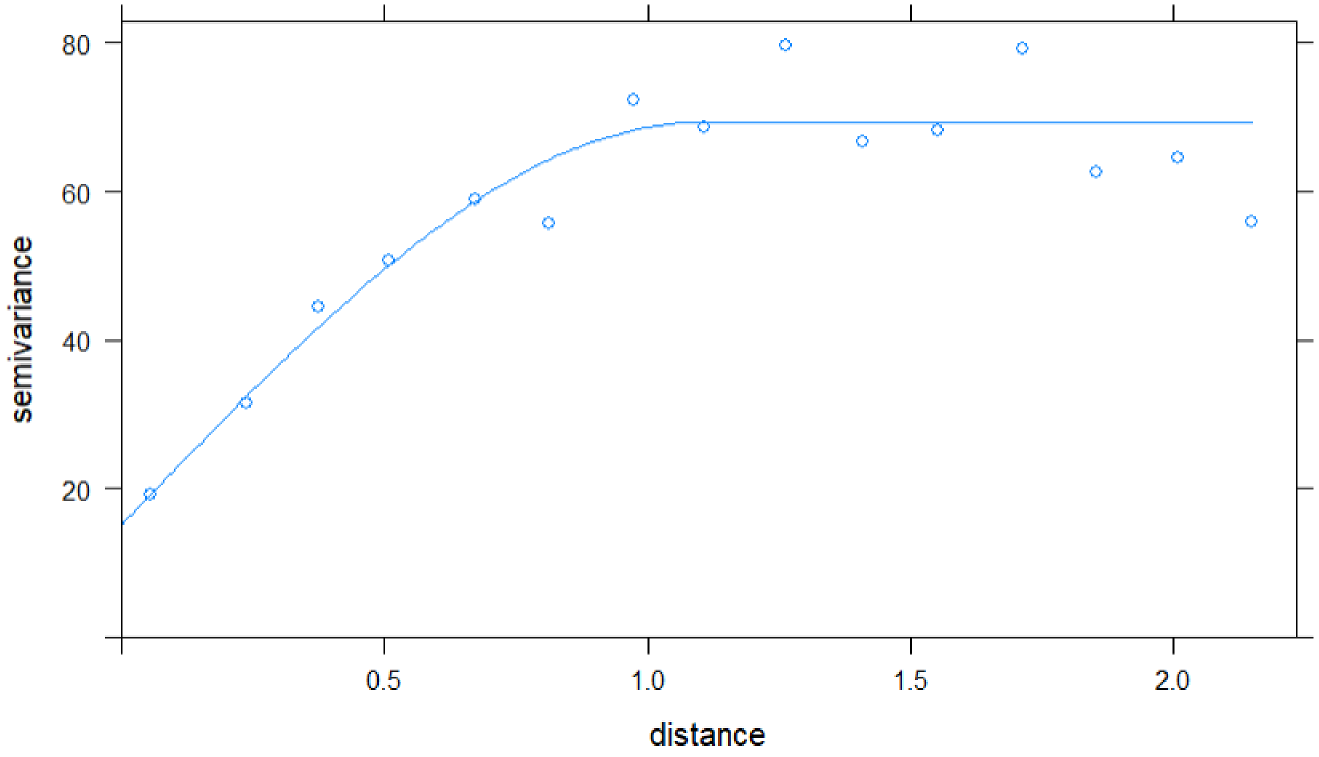
Let’s do it in R: Fit the variogram

1. Convert to `sp` objects:

```
jura.pred.df <- jura.pred
coordinates(jura.pred) <- ~Xloc + Yloc
jura.val.df <- jura.val
coordinates(jura.val.df) <- ~Xloc + Yloc
jura.grid.df <- jura.grid
coordinates(jura.grid.df) <- ~Xloc + Yloc
```

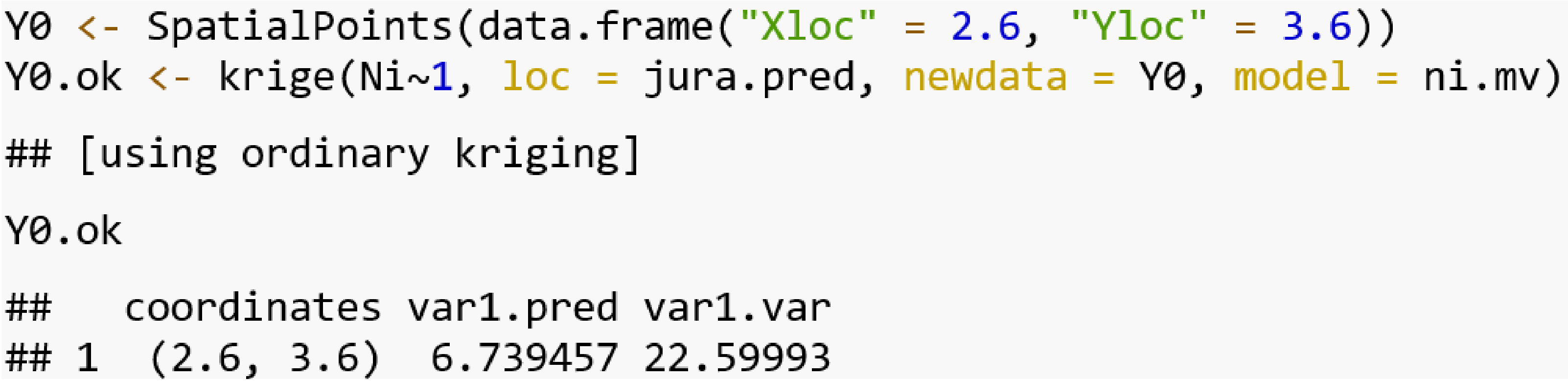
2. Fit the variogram model:

```
ni.ev <- variogram(Ni~1, data = jura.pred)
ni.mv <- fit.variogram(ni.ev, model = vgm(70, "Sph", 1, 15), fit.method = 7)
plot(ni.ev, ni.mv)
```



Let's do it in R: Prediction of one point

- Predict the Nickel concentration at one unsampled location u_0 with co-ordinates (2.6, 3.6) (on the local co-ordinate system):



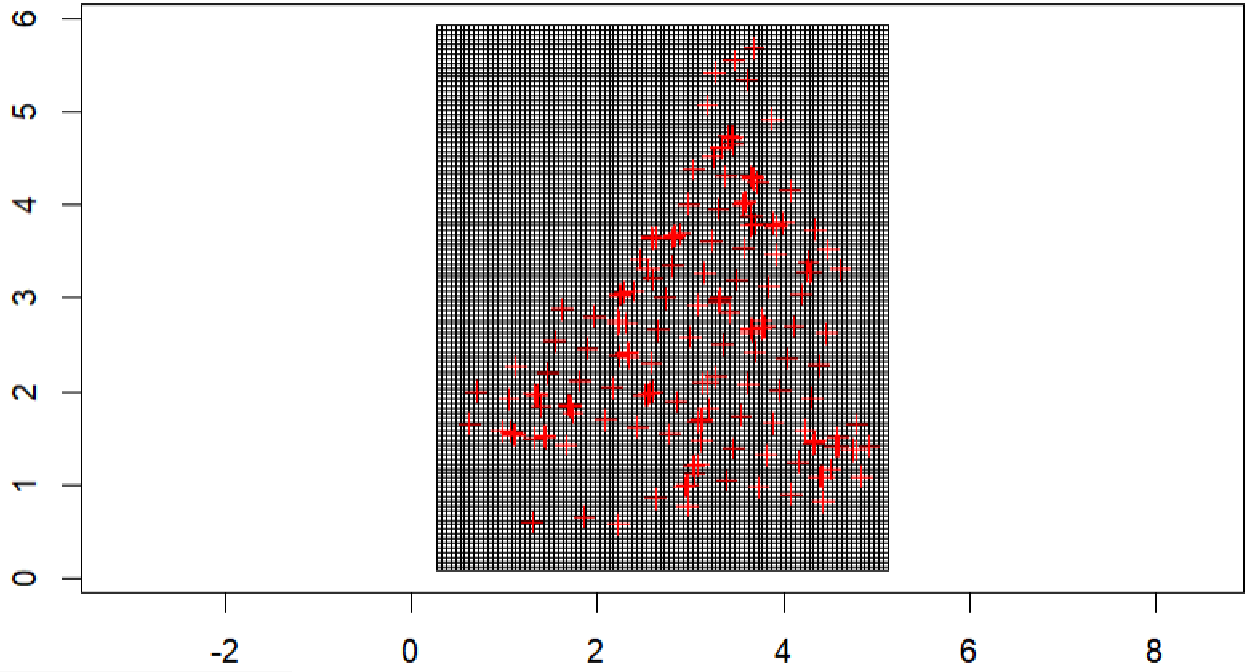
```
Y0 <- SpatialPoints(data.frame("Xloc" = 2.6, "Yloc" = 3.6))
Y0.ok <- krige(Ni~1, loc = jura.pred, newdata = Y0, model = ni.mv)

## [using ordinary kriging]

Y0.ok

##   coordinates var1.pred var1.var
## 1  (2.6, 3.6)  6.739457 22.59993
```

- `var1.pred` is the kriging predictor at location u_0
- `var1.var` is the kriging variance at location u_0

Let’s do it in R: Prediction of a surface

- Ordinary Kriging on a regular grid of unsampled locations (create a regular grid covering approximately the same area as the measurements with 50m spacing):

```
grid <- expand.grid(Xloc = seq(0.3, 5.1, by=0.05), Yloc = seq(0.1, 5.9,
by=0.05))
grid <- SpatialPoints(grid)
gridded(grid) <- TRUE
head(grid)

## Object of class SpatialPixels
## Grid topology:
##       cellcentre.offset cellsize cells.dim
## Xloc                0.3     0.05        97
## Yloc                0.1     0.05       117
## SpatialPoints:
##       Xloc Yloc
## [1,] 0.30  0.1
## [2,] 0.35  0.1
## [3,] 0.40  0.1
## [4,] 0.45  0.1
## [5,] 0.50  0.1
## [6,] 0.55  0.1
## Coordinate Reference System (CRS) arguments: NA

plot(grid, axes = TRUE)
plot(jura.pred, col = "red", add = TRUE)#Add the locations of the known
measurements
```



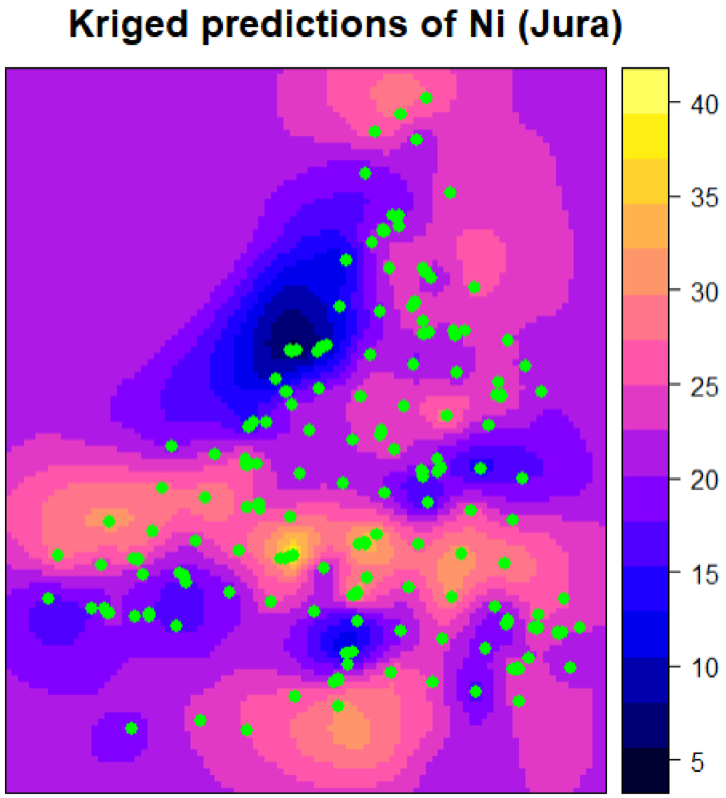
Let's do it in R: Prediction of a surface

Implement Kriging:

```
jura.ok <- krige(Ni~1, loc = jura.pred, newdata = grid, model = ni.mv)

## [using ordinary kriging]
```

plot the predicted results (notice the prediction values tend to the spatial mean as we go further from the sample points):

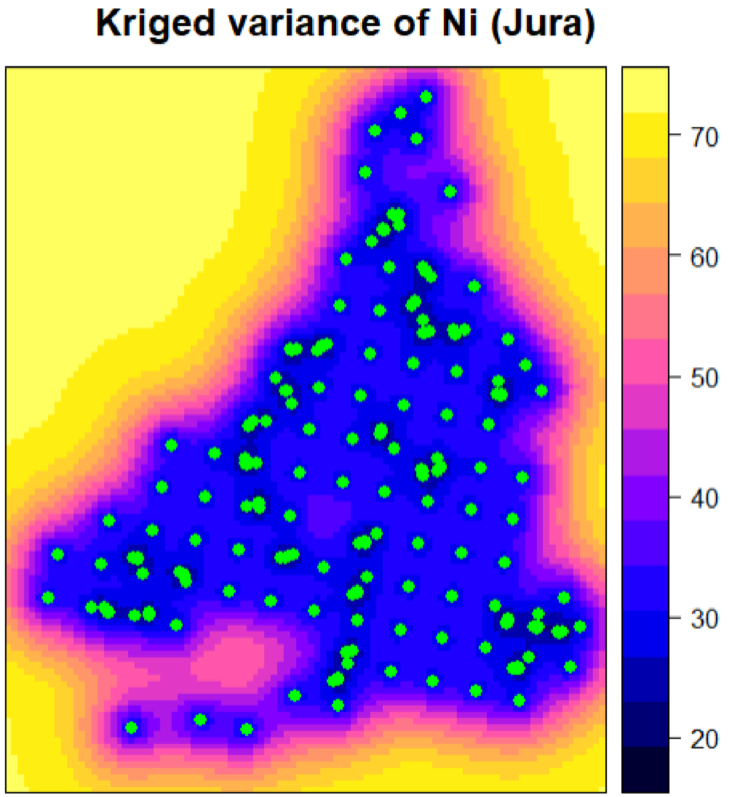
```
spplot(jura.ok, "var1.pred", sp.layout=
          list("sp.points", pch = 19, col = "green", jura.pred),
       main="Kriged predictions of Ni (Jura)")
```



Let’s do it in R: Prediction of a surface

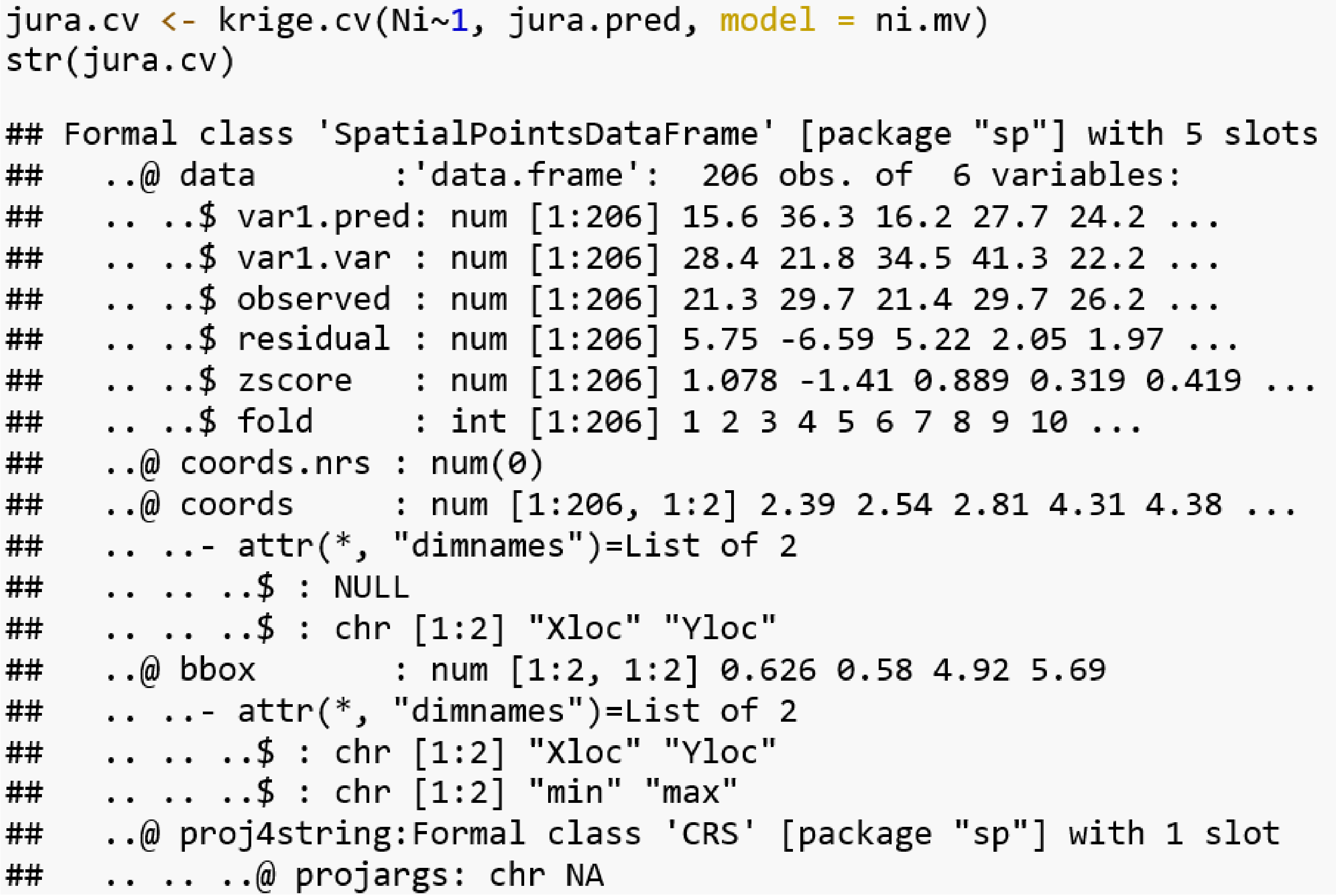
Plot the variance of predicted results (notice that the variance increases as we go further from the sample points)

```
spplot(jura.ok, "var1.var", sp.layout= list("sp.points", pch = 19, col =
"green", jura.pred), main="Kriged variance of Ni (Jura)")
```



Let’s do it in R: Accuracy measures

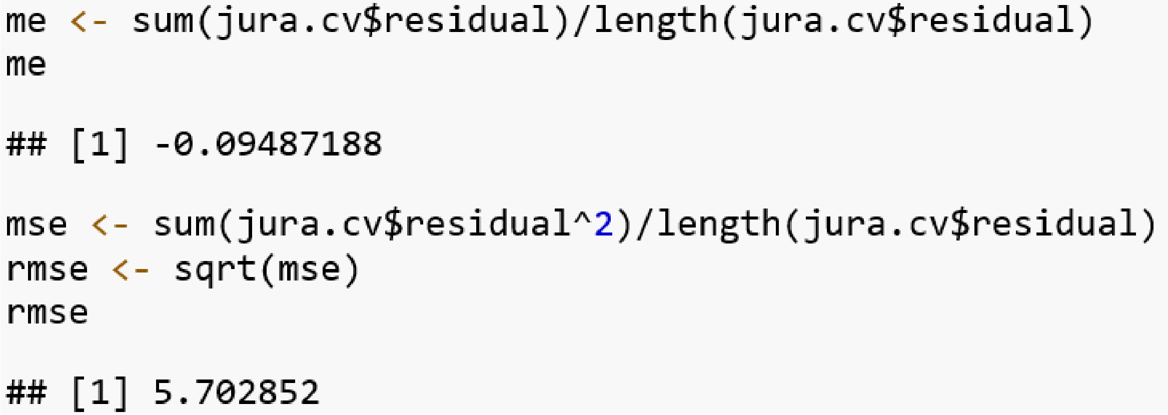
- Cross-validation to assess the bias and accuracy:



```
jura.cv <- krige.cv(Ni~1, jura.pred, model = ni.mv)
str(jura.cv)

## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data       :'data.frame':  206 obs. of  6 variables:
##   .. ..$ var1.pred: num [1:206] 15.6 36.3 16.2 27.7 24.2 ...
##   .. ..$ var1.var : num [1:206] 28.4 21.8 34.5 41.3 22.2 ...
##   .. ..$ observed : num [1:206] 21.3 29.7 21.4 29.7 26.2 ...
##   .. ..$ residual : num [1:206] 5.75 -6.59 5.22 2.05 1.97 ...
##   .. ..$ zscore   : num [1:206] 1.078 -1.41 0.889 0.319 0.419 ...
##   .. ..$ fold     : int [1:206] 1 2 3 4 5 6 7 8 9 10 ...
##   ..@ coords.nrs : num(0)
##   ..@ coords     : num [1:206, 1:2] 2.39 2.54 2.81 4.31 4.38 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:2] "Xloc" "Yloc"
##   ..@ bbox       : num [1:2, 1:2] 0.626 0.58 4.92 5.69
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "Xloc" "Yloc"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr NA
```

Let’s do it in R: Accuracy measures

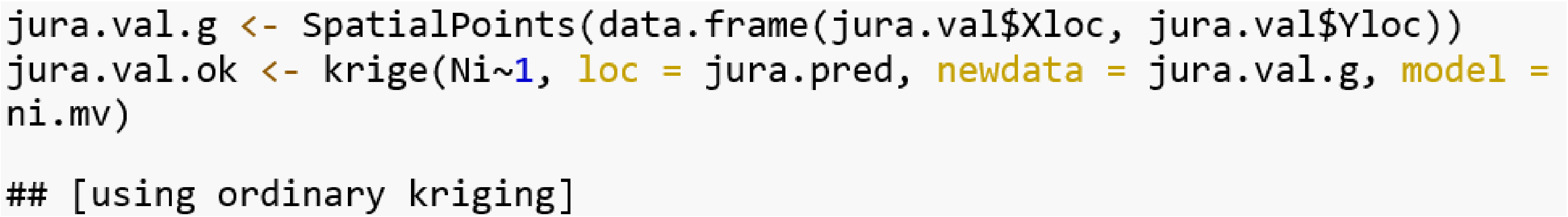
- Bias (mean error):



```
me <- sum(jura.cv$residual)/length(jura.cv$residual)
me

## [1] -0.09487188

mse <- sum(jura.cv$residual^2)/length(jura.cv$residual)
rmse <- sqrt(mse)
rmse

## [1] 5.702852
```

Let’s do it in R: Different models

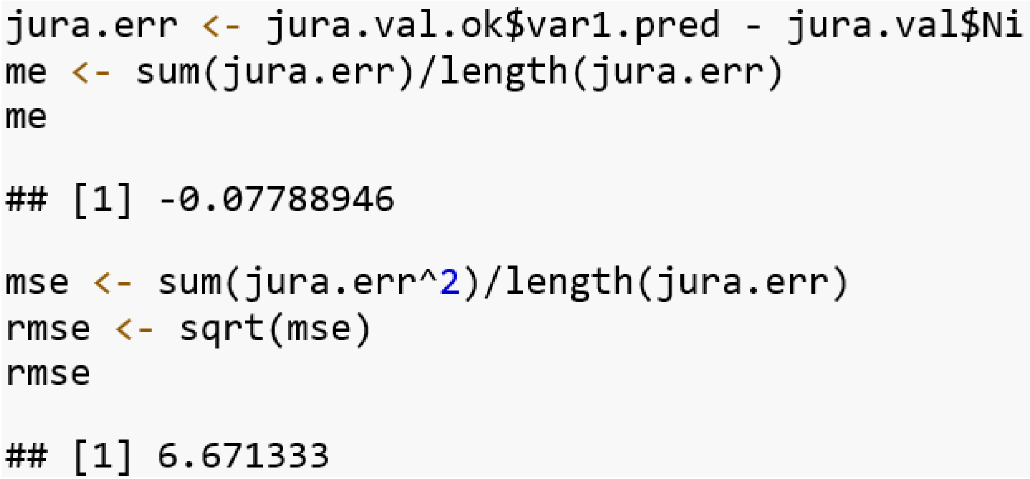
- Assess the impact of different variogram models using *me* and *rmse*.
- Assess the predictions against separate data sets: Split the data set in two - estimate with one, assess with the other. The `jura.val` set is provided for this purpose. Perform Kriging on `jura.val`.



```
jura.val.g <- SpatialPoints(data.frame(jura.val$Xloc, jura.val$Yloc))
jura.val.ok <- krige(Ni~1, loc = jura.pred, newdata = jura.val.g, model =
ni.mv)

## [using ordinary kriging]
```

Let’s do it in R: Different models

- Calculate the error, *me* and *rmse*:



```
jura.err <- jura.val.ok$var1.pred - jura.val$Ni
me <- sum(jura.err)/length(jura.err)
me

## [1] -0.07788946

mse <- sum(jura.err^2)/length(jura.err)
rmse <- sqrt(mse)
rmse

## [1] 6.671333
```

Links and resources

- Some links to understand Kriging:
  - Columbia University: https://www.publichealth.columbia.edu/research/population-health-methods/kriging-interpolation
  - Kriging on Wikipedia: https://en.wikipedia.org/wiki/Kriging
  - ArcGIS: https://desktop.arcgis.com/en/arcmap/10.3/tools/3d-analyst-toolbox/how-kriging-works.htm
- History of Danie Krige: https://www.gislounge.com/danie-krige-kriging/
- Textbook explanation of Kriging: Diggle and Ribeirro, Model-based Geostatistics, Springer, New York, 2007 (Chapters 6.2, 6.2, 6.3)