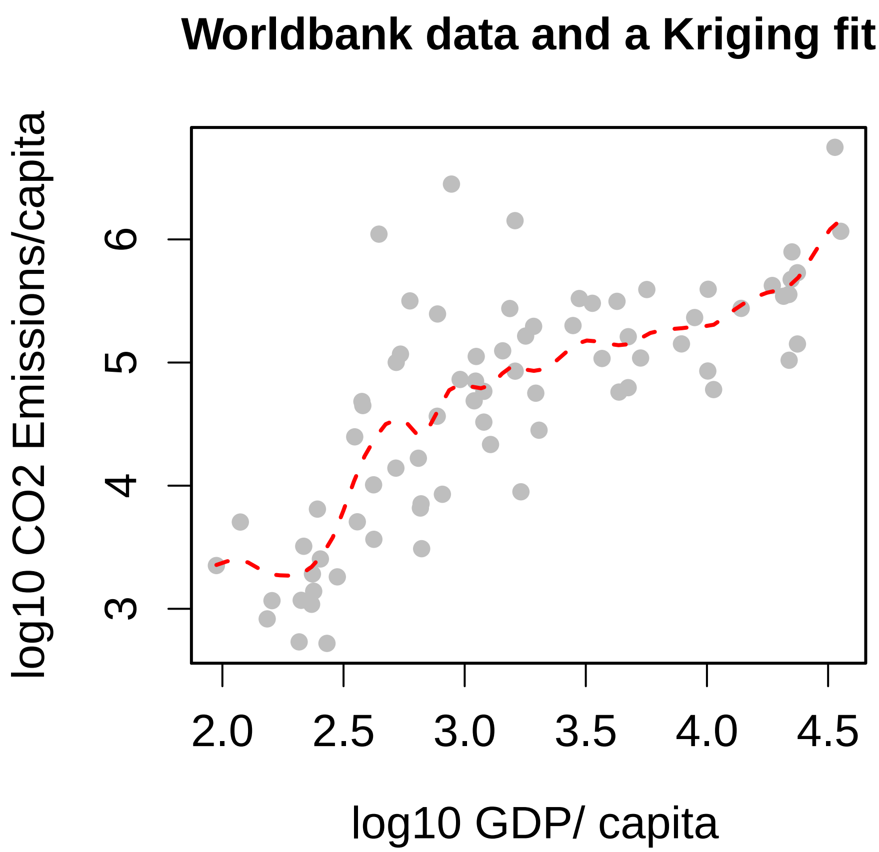
Part 1 Lab Day 2

ISI Short Course | June 1-3

Creating radial basis functions (RBF)

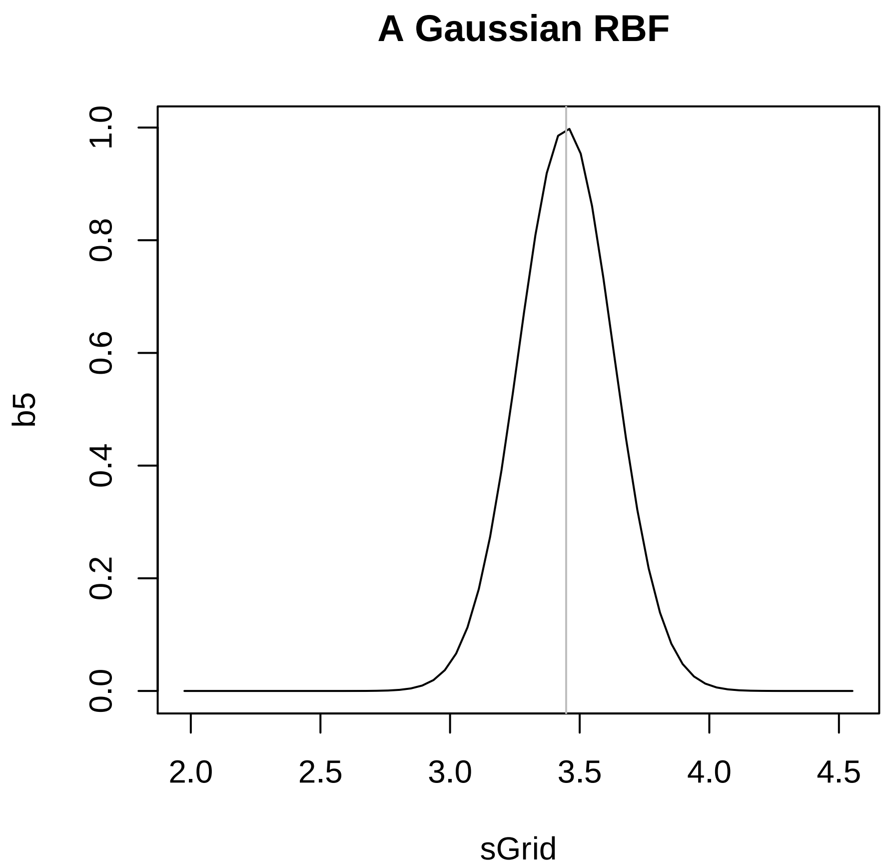
In 1-d distance is just absolute value. Here is a series of differnt ways to code this. Using the scaled Gaussian function as the bump. We will use the range of the World Bank CO2 data set for these examples

```
# set up data
data("WorldBankCO2")
# log10 CO2 emissions by country
x<- as.matrix( log10( WorldBankCO2[,"GDP.cap"]))
y<- log10( WorldBankCO2[,"CO2.cap"])
sGrid <- seq( min(x), max(x), length.out = 60)
fields.style()
plot( x, y, xlab=" log10 GDP/ capita", ylab="log10 CO2 Emissions/capita",
      pch=16, col="grey")
fit<- spatialProcess( x,y)
lines( sGrid, predict(fit,sGrid), col="red", lwd=2, lty=2)
title( "Worldbank data and a Kriging fit")
```



```r
u <- seq( min(x), max(x), length.out=8)

# a single basis function  -- the 5th one
b5<- rep(NA, 60)
 delta<- max(x) - min( x)
for( k in 1:60){
  d5<- abs(sGrid[k]- u[5])/(.1*delta)
  b5[k]<-  exp( - d5^2)
}
#
plot( sGrid, b5, type="l")
xline( u[5], col="grey")
title(" A Gaussian RBF")
```



For loops are to be avoided in R if there are simpler functions Here is a better way to code this

```r
dVec<- rdist( sGrid, u[5])/(.1*delta)
b5<- exp( - dVec^2 )
```

Typically we want to gt all the RBFs for fitting and evaluation so the final coding step is to generate the whole basis matrix that is **60X10**. In this result column 5 (**basisMatrix[,5]**) is the same as b5 above

```r
bigD<- rdist( sGrid, u)/(.1*delta)
basisMatrix1<- exp( - bigD^2)
dim( basisMatrix1)
```

```
## [1] 60  8
```

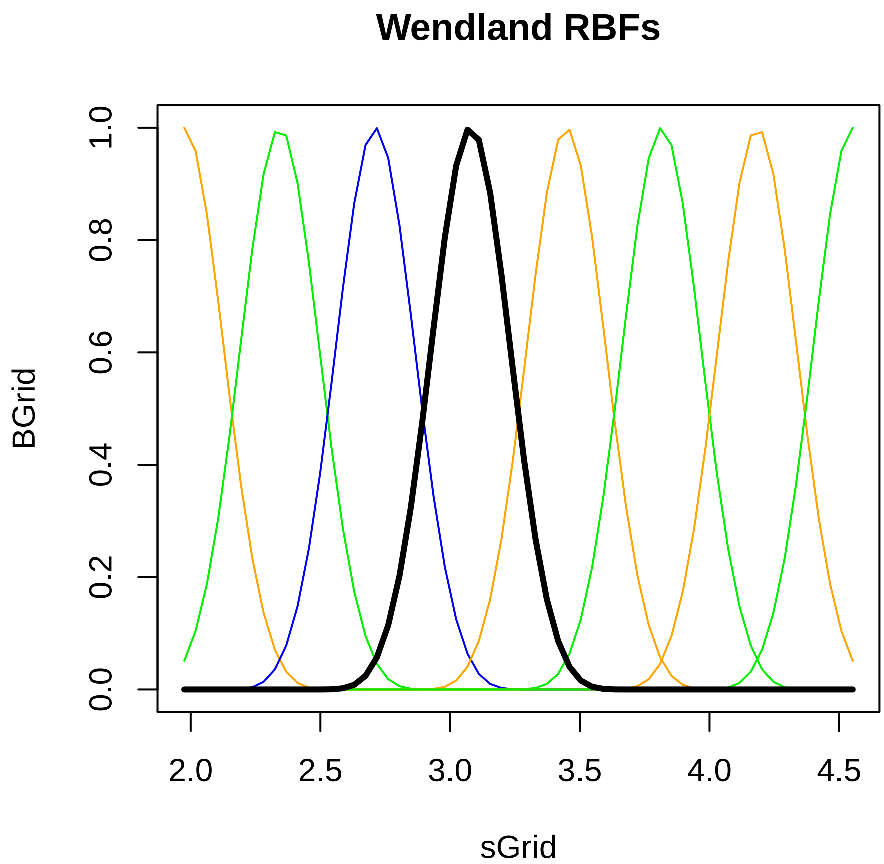
Finally, the Wendland shape is better for computing because it is zero beyond a certain range. Increase the scale so there is more overlap.

```
BGrid<- WendlandFunction(rdist( sGrid, u)/(2*delta/8))
dim( BGrid)
```

```
## [1] 60  8
```

And a plot to see all these guys . . .

```
matplot( sGrid, BGrid, type="l", lty=1)
lines( sGrid, BGrid[,4], col="black", lwd=3)
title( "Wendland RBFs")
```



Sometimes we also want to include a constant and linear term. Here X is ready for curve fitting to the World Bank data. Redo some of the computations to make it easy to change number of basis functions.

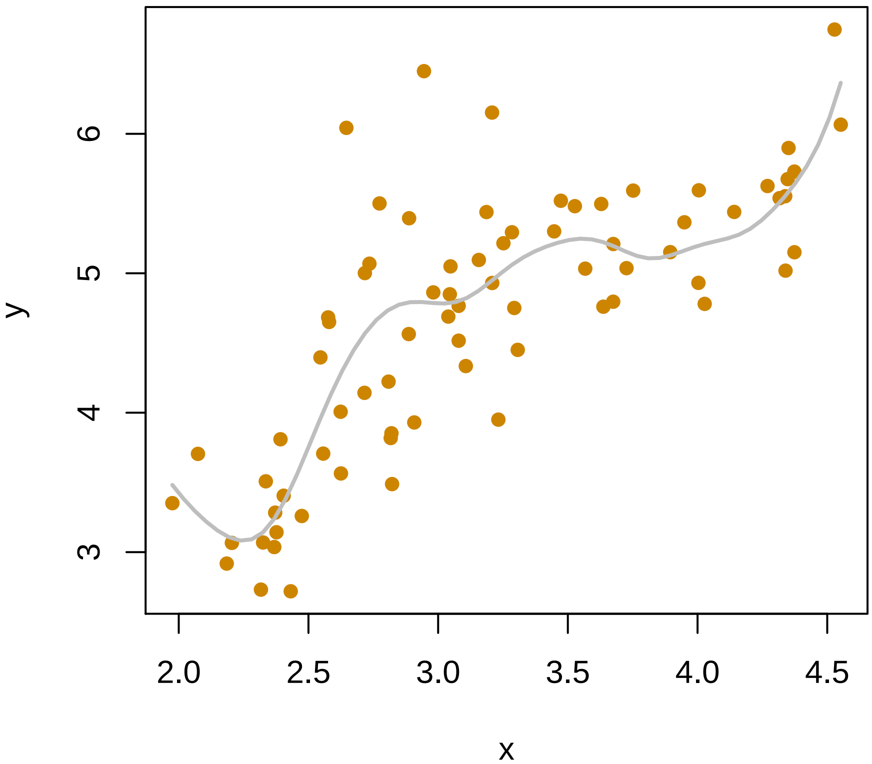
```
u <- seq( min(x), max(x), length.out=8)
BGrid<- WendlandFunction(rdist( sGrid, u)/(3*delta/8))
XGrid<- cbind( 1, sGrid, BGrid )
XB<- WendlandFunction( rdist( x, u)/( 3*(delta/8) ) )
X<- cbind( 1, x, XB)
dim( X)
```

```
## [1] 75 10
```

```
# fit the data by OLS
fitRBF<- lm( y~ X - 1 )
#print( summary( fitRBF))
```

Take a look at the fitted curve.

```
plot( x, y, col="orange3", pch=16)
gHat<- XGrid%*% fitRBF$coefficients
lines(sGrid, gHat, lwd=2, col="grey" )
```



Exercises

1. What happens to the overlap in the Wendland basis functions if the 3 in the scaling is decreased to 1.5?
2. The example fit is probably too "wiggly". Find a comnbination of overlap and number of basis functions that visually is a better fit to these data.