Forecasting with Temporal and Spatial Data in R

The following is a script file (RCode/RmodelDependencies.R) containing all R code of all sections in this slide set.

Time Series Modeling

```
data(ice.river,package='tseries')
head(ice.river)
decomp <- stl(ice.river[,1], s.window=10)
```

```
plot(decomp)
```

```
ir <- xts(ice.river[,1],
          seq(ymd('1972-01-01'),
              by='day',
              len=nrow(ice.river)))
plot(ir)
```

```
(s <- ir[1:10])
diff(s)
```

```
diff(s,diff=2)
log10(s)
```

```
plot(ir)
acf(ir)
```

```
library(forecast)
s <- window(ice.river[,"temp"],start=1972,end=1973)
plot(s, main="A Simple Moving Average")
lines(ma(s,order=20,centre=FALSE),col="red")
```

```
library(forecast)
s <- window(ice.river[,"temp"],start=1972,end=1973)
model <- ses(s, alpha=0.3, initial="simple")
plot(s, main="An Exponential Moving Average Model")
lines(fitted(model),col="red")
```

```
forecast(model,h=5)
plot(forecast(model,h=5))
```

```
hw <-  HoltWinters(ice.river[,"temp"])
plot(hw)
```

```
forecast(hw,h=5)
plot(forecast(hw,h=5))
```

```
a1 <- Arima(s,order=c(1,1,2))
a1
```

```
plot(s, main="An ARIMA(1,1,2) Model")
lines(fitted(a1),col="red")
```

```
a2 <- auto.arima(s)
a2
```

```
plot(s, main="An Automatically Selected ARIMA Model")
lines(fitted(a2),col="red")
```

```
forecast(a2,h=5)
plot(forecast(a2,h=5))
```

```
library(DMwR2)
dat <- createEmbedDS(ice.river[,"temp"], emb = 6)
head(dat)
head(ice.river[,"temp"])
```

```
library(DMwR2)
library(rpart.plot)
tr <- rpartXse(T ~ ., as.data.frame(dat))
prp(tr)
```

Spatial Data Modeling

```
library(sp)
data(meuse)
coordinates(meuse) <- ~x+y
proj4string(meuse) <- CRS("+init=epsg:28992")
meuse[1:2,]
```

```
distsTo1 <- spDistsN1(meuse, meuse[1,]) # Euclidean
nn3 <- meuse[order(distsTo1)[2:4],]
meuse[1,]
nn3
distsTo1GC <- spDistsN1(meuse, meuse[1,], longlat=TRUE) # Great Circle
```

```
library(automap) # extra package you need to install
kr <- autoKrige(zinc ~ 1, meuse)
plot(kr)
```

```
data(meuse.grid)
gridded(meuse.grid) <- ~x+y
kr2 <- autoKrige(zinc ~ 1, meuse, meuse.grid)
plot(kr2)
```

```
getVars <- function(location, data, var, nns=c(3,5,10), funcs=c("mean","var")) {
    dists <- spDistsN1(data,location)
    o <- order(dists)
    res <- lapply(nns, function(nn) {
        ns <- data[o[1:nn],]
        vals <- ns[[var]]
        nms <- paste(var,funcs,nn,sep=".")
        vs <- sapply(funcs,function(f) do.call(f,list(vals)))
        names(vs) <- nms
        vs
    })
    unlist(res)
}
getVars(meuse.grid[1,],meuse,"zinc")
```

```
set.seed(1234)
traindat <- NULL
for(r in 1:nrow(meuse)) traindat <- rbind(traindat,getVars(meuse[r,],meuse[-r,],"zinc"
))
traindat <- data.frame(traindat,tgtZinc=meuse[["zinc"]])
head(traindat)
```

```
set.seed(1234)
szTest <- 100
testPoints <- meuse.grid[sample(1:nrow(meuse.grid),szTest),]
dat <- NULL
for(r in 1:szTest) dat <- rbind(dat,getVars(testPoints[r,],meuse,"zinc"))
head(dat)
```

```
library(e1071)
s <- svm(tgtZinc ~ ., traindat, cost=10, epsilon=0.1)
ps <- predict(s,dat)
head(ps)
```