# Predictions using \texttt{lm} and spatial predictions

APPM Methods and Analysis of Large Data Sets

\begin{verbatim}
#
library(dataWorkshop)

## Loading required package: maps
## Loading required package: fields
## Loading required package: spam
## Loading required package: grid
## Spam version 0.41-0 (2014-02-26) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
## The following objects are masked from 'package:base':
##
##   backsolve, forwardsolve

# first some review
# predictions for PilotSUV data
# Here is a basic model. This may not be the best but a reasonable
# one.
   data(PilotSUV)
   fitSUV<-	exttt{lm}( price \sim \texttt{model} + \texttt{year} + \texttt{mileage}, \texttt{data} = \texttt{PilotSUV})
# check the fit
   fields.style()
   set.panel(2,1)

## plot window will lay out plots in a 2 by 1 matrix

plot( predict( fitSUV), residuals( fitSUV))
yline(0, col=1)
indBuy<- \texttt{which.min}( residuals( fitSUV))

qqnorm( residuals( fitSUV))
\end{verbatim}
# inference for predicted values

obj <- predict(fitSUV, se.fit = TRUE)

# summary of predicted values for the 20 most negative residuals

set.panel()  

## plot window will lay out plots in a 1 by 1 matrix

CI <- cbind(obj$fit - 2 * obj$se.fit, obj$fit, obj$fit + 2 * obj$se.fit) 
iorder <- order(fitSUV$residuals) 
N <- length(PilotSUV$price) 
temp <- cbind(PilotSUV$price, CI)
# considering prediction intervals
obj2 <- predict(fitSUV, interval="prediction", newdata = PilotSUV)
CI2 <- obj2[, c(2, 1, 3)]
iorder <- order(fitSUV$residuals)
N <- length(PilotSUV$price)
temp <- chind(PilotSUV$price, CI2)
temp <- temp[iorder,]
matplot( temp[1:20,], 1:20, type="p", 
pch = c("o", "|", "+", "|"), cex = 1.2, col = c(1, 2, 3, 2),
ylab = "Rank by residual", xlab = "Asking Price")
yline( 1:20, lwd = .5, col = "grey")
# getting standard errors for a spatial prediction
#
# subset of North American Rainfall

```r
data( NorthAmericanRainfall)
x<- cbind( NorthAmericanRainfall$longitude, NorthAmericanRainfall$latitude)
y<- NorthAmericanRainfall$precip
# change to inches of rainfall
y<- y/254
# select out the subset of locations from reat Plains
ind <- (x[,1] <= -90) & (x[,1]>= -103) & (x[,2] >= 32) & (x[,2] <= 47)
x<- x[ind,]
y<- y[ind]
```
par.grid <- list(theta = exp(seq(log(1), log(10), 30)))

# explore likelihood with smoothness 1.0 along with lambda and theta
# NOTE: this function automatically optimizes over lambda so saves that grid search.

MLEObj1 <- mKrig.MLE(x, y, Covariance = "Matern", cov.args = list(smoothness = 1.0),
                      par.grid = par.grid, lambda = rep(1, 30), verbose = TRUE)

theta.MLE <- MLEObj1$cov.args.MLE$theta
lambda.MLE <- MLEObj1$lambda.MLE

# check results
plot(par.grid$theta, MLEObj1$summary[, "lnProfLike"],
     ylab = "profile log Likelihood", xlab = "theta")
xline(theta.MLE, col = 2, lwd = 2)
fitRainfall <- mKr["m", y, Covariance="Matern", cov.args=list(smoothness=1.0, theta=theta.MLE), lambda=lambda.MLE)

fitSurface <- predictSurface(fitRainfall)
SESurface <- predictSurfaceSE(fitRainfall)

# take a look
set.panel(2,1)

## plot window will lay out plots in a 2 by 1 matrix

image.plot(fitSurface, axes=FALSE)
US(add=TRUE, lwd=2, col="grey")
image.plot(SESurface, axes=FALSE)
US(add=TRUE, lwd=2, col="grey")
points(x, col="white", cex=.5, pch=16)
# less that 9 inches of rain
set.panel()

## plot window will lay out plots in a 1 by 1 matrix

ind9<- fitSurface$z < 9 - 2*SESurface$z
image( fitSurface$x, fitSurface$y, ind9, col=c("white", "grey") )
contour( fitSurface, level=c( 8,9 ), add=TRUE)
# NOTE: Tricky to interpret what this plot means because significance is found across many grid boxes

# a better way to make this inference
#(add timing just to have reference as to how long this takes)

```r
system.time(
  simObj <- sim.mKrig.approx( fitRainfall, M=10, nx=50, ny=50, gridExpansion= 3 )
)
```

## user system elapsed
## 19.604  0.716  20.159

```r
set.panel()
```

## plot window will lay out plots in a 1 by 1 matrix
# in class bootstrap example
NBoot<- 1000
hold<- matrix( NA, nrow=194, ncol=NBoot)  
sigma<- (summary(fitSUV)$sigma)
for( k in 1: NBoot)  
  {  
  }
cat( k, " ")#
yFake <- predict( fitSUV ) + rnorm(194, sd=sigma )#
obj <- lm( yFake ~ model + year + mileage, data = PilotSUV )#
hold[,k] <- 100*residuals(obj)/predict(obj)#
}

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 ...
##
## SEpercentage <- apply( hold, 1, FUN="sd")
hist( SEpercentage)
PilotSUV[170,]

## year model price mileage SUV color distance
## 175 2005 HEX 21.9 19 SUV 678 2004

100*residuals(fitSUV)[170]/ predict( fitSUV)[170]

## 175
## -23.71

# 95% CI from bootstrap sample
quantile( hold[170,],c(.025, .975))

## 2.5% 97.5%
## -12.90 13.32

# compare to approx 95% CI bounds based on standard deviation of bootstrap results
2* SEpercentage[170]

## [1] 13.55