STAT 506 Midterm Exam

Steve Harms
March 5, 2018

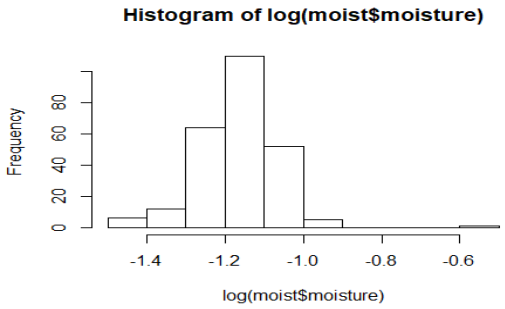
3)

a)

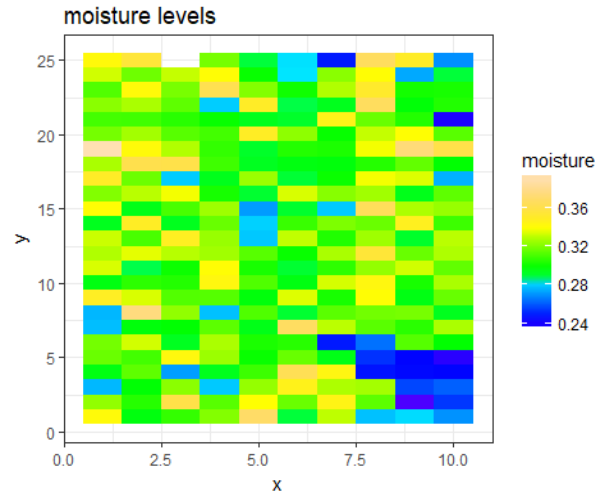
I use a log transformation of the moisture levels, although the model fits approximately the same without it.

```
moist<- read.table("moisture.txt", header = T)
names(moist) <- c("x", "y", "density", "moisture")
#Create a variable with log transformation (may or may not be needed, but I did it anyway)
moist$logm <- log(moist$moisture)
#Another dataset with what appears to be an outlier removed, just in case
moist.rm <- moist[-75,]
#A quick histogram shows that it's somewhat normal

hist(log(moist$moisture))
```



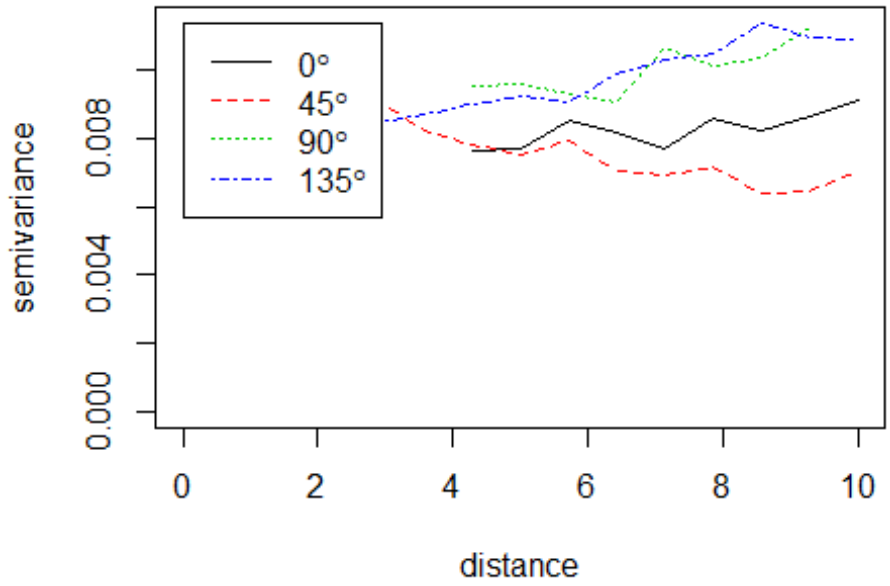

```
#An image of our spatial data points
gg <- ggplot(data = moist.rm, aes(x=x, y=y))
ii <- gg + geom_tile(mapping = aes(fill = moisture)) +  scale_fill_gradientn(colours = topo.colors(250))+
  theme_bw() + xlab("x") + ylab("y")+ ggtitle("moisture levels")
ii
```



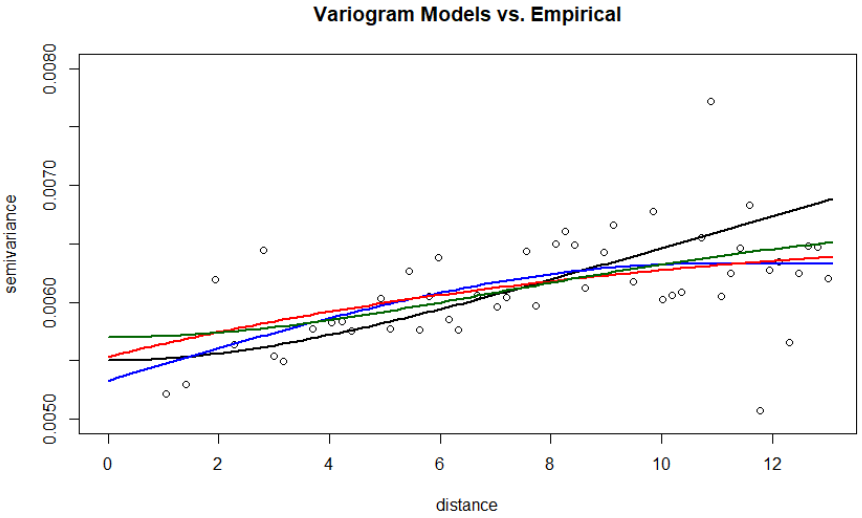

```
#create geodata sets
moistgeo <- as.geodata(moist, coords.col = 1:2, data.col = 5, covar.col = 3)
moistgeo.rm <- as.geodata(moist.rm, coords.col = 1:2, data.col = 5, covar.col = 3)
#estimate empirical variogram
omndvar <- variog(moistgeo, trend = ~density, option = "bin", uvec = seq(0, 13, length = 75 ))
```

A quick look at the directional variogram shows that there isn't much concern about anisotropy, although it's not as clear as in other examples. The variograms diverge a bit at reasonably short lags, but for now I will just assume isotropy.

```
var.direct <- variog4(moistgeo, uvec = seq(0, 10, length = 15 ), direction = c(0,pi/4,pi/2,3*pi/4))
plot(var.direct)
```




```
#some candidate models
vario.exp <- variofit(omndvar, cov.model = "exp", weights = "cressie", nugget = .0048, fix.nugget=F,
ini.cov.pars = c(.001, 10))
vario.sph <- variofit(omndvar, cov.model = "spherical", weights = "cressie", nugget = .0048,
fix.nugget=F,ini.cov.pars = c(.002, 10))
vario.gauss <- variofit(omndvar, cov.model = "gaussian", weights = "cressie", nugget = .005,
fix.nugget=F,ini.cov.pars = c(.002, 10))
vario.matern <- variofit(omndvar, cov.model = "matern", weights = "cressie",nugget = .0055, fix.nugget=F,
kappa = .75, fix.kappa=F)
plot(omndvar, main = "Variogram Models vs. Empirical") + lines(vario.matern) + lines(vario.sph, col = "blue")
+ lines(vario.exp, col = "red") + lines(vario.gauss, col = "dark green")
```



It looks like the spherical model in blue fits best, so I will use that one to fit in the next part. The exponential model is not too far off, so that one could be used as well.

b)

One interesting part here is that it appears there's a column effect (i.e., a trend in the x-coordinates), but the kriging predictions don't have much of an effect. I include a quick image of the predictions with this trend but don't use it in the next part.

```
#fit models with covariates
sph.fit.nt = likfit(moistgeo, trend= ~moistgeo$covariate[,1], fix.nugget = F, nugget =.0048, ini =
c(.002,10),
                    cov.model = "spherical", lik.method = "REML")
sph.fit.nc = likfit(moistgeo, fix.nugget = F, nugget =.0048, ini = c(.002,10),
                    cov.model = "spherical", lik.method = "REML")
```

```
sph.fit.rm = likfit(moistgeo.rm, trend= ~moistgeo.rm$covariate[,1], fix.nugget = F, nugget =.0048, ini =
c(.002,10),
                    cov.model = "spherical", lik.method = "REML")
```

For our model with covariate, I get the following REML estimates for nugget, range, and partial sill:

```
sph.fit.nt$nugget
## [1] 0.005347021
sph.fit.nt$phi
## [1] 9.405413
sph.fit.nt$sigmasq
## [1] 0.002256946
```

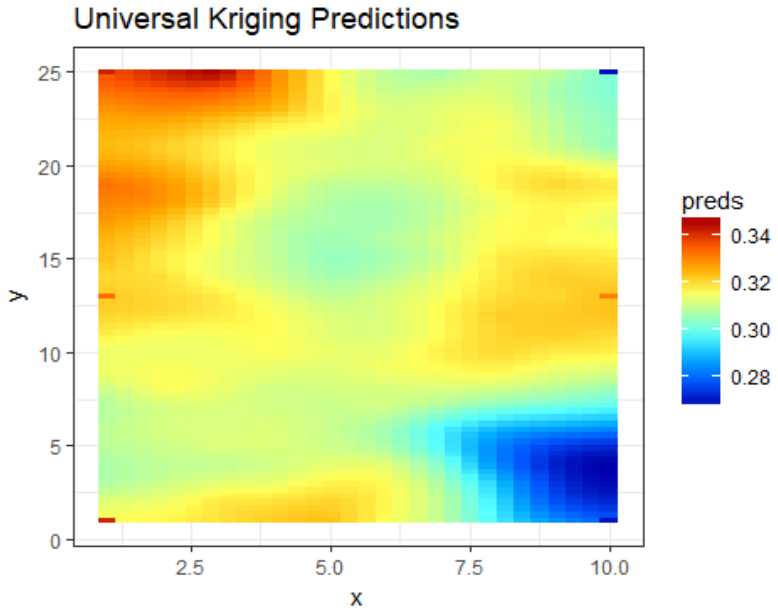
c)

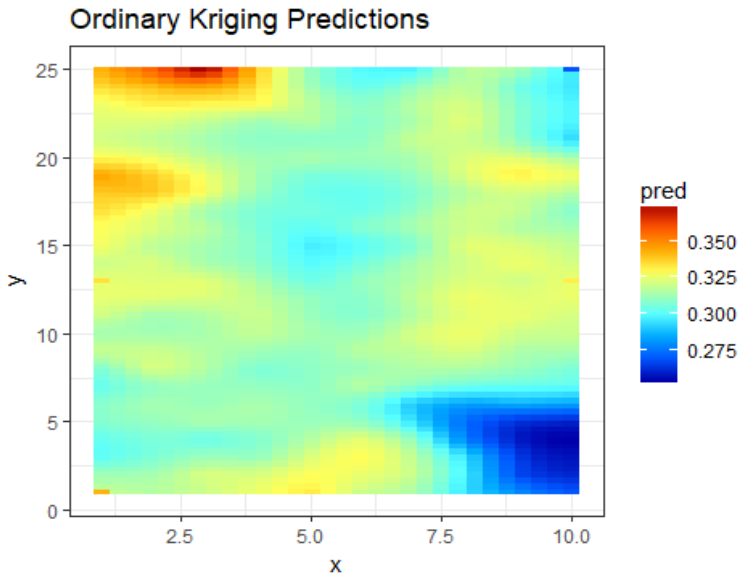
```
#test for covariate significance in our model
trend.coef = sph.fit.nt$beta
names(trend.coef) = c("Intercept","density")
trend.coef
##  Intercept    density
## -0.3044718 -0.5787528
trend.cov = sph.fit.nt$beta.var
trend.cov
##              V1           V2
## V1  0.008299747 -0.005373007
## V2 -0.005373007  0.003610706
t.density = trend.coef[2]/sqrt(trend.cov[2,2])
cat("t-value =",t.density,"\n")
## t-value = -9.631568
p.density = 2*(1 - pnorm(abs(t.density)))
cat("p-value =",p.density,"\n")
## p-value = 0
```

We can conclude from the small p-value that the covariate “density” has a significant effect when the spatial adjustment is included in the model.

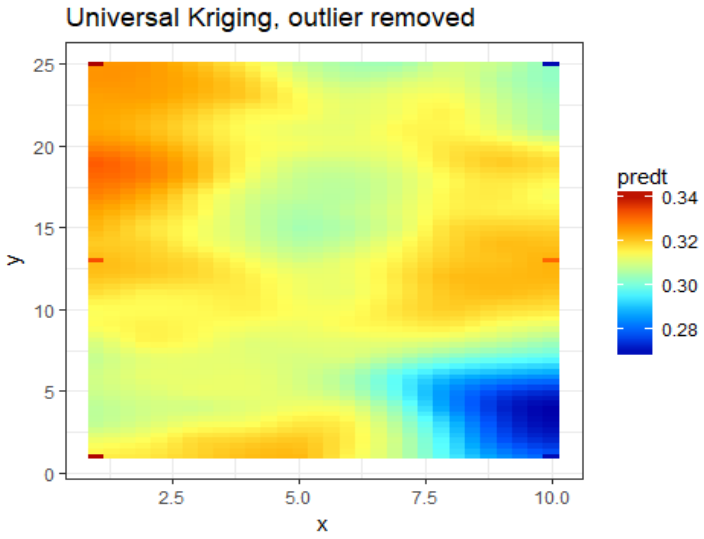
d)

```
xrange <- range(moist[,1])
yrange <- range(moist[,2])
grid <- expand.grid(x = seq(xrange[1],xrange[2],l=30),y=seq(yrange[1],yrange[2],l=75))
#with covariate but no trend
kc.sph.nt <- krige.conv(geodata=moistgeo, locations= grid, krige=krige.control(type.krige = "OK", obj.model =
sph.fit.nt))
krigimage.nt <- data.frame(x=grid$x,y = grid$y, preds=exp(kc.sph.nt$predict))
aa <- ggplot(data=krigimage.nt,mapping = aes(x = x, y = y))
cc<- aa + geom_tile(mapping = aes(fill = preds)) +  scale_fill_gradientn(colours = matlab.like(80))+
  theme_bw() + xlab("x") + ylab("y") + ggtitle("Universal Kriging Predictions")
cc
#without trend or covariate
kc.sph <- krige.conv(geodata=moistgeo, locations= grid,krige=krige.control(type.krige="OK",obj.model =
sph.fit.nc))
krigimage <- data.frame(x=grid$x,y = grid$y, pred=exp(kc.sph$predict))
q <- ggplot(krigimage,mapping = aes(x = x, y = y))
u <- q + geom_tile(mapping = aes(fill = pred)) +  scale_fill_gradientn(colours = matlab.like(80))+
  theme_bw() + xlab("x") + ylab("y") + ggtitle("Ordinary Kriging Predictions")
u
```





A quick look at what it would look like if we removed the outlier from the analysis, not much different:



Finally, a report of the mean square differences as a percentage of total sd in the data:

```
msd <- mean((exp(kc.sph$predict) - exp(kc.sph.nt$predict))^2)
msd
## [1] 1.980247e-05
```

```
100*msd/sd(moist$moisture)
## [1] 0.06173717
100*msd/var(moist$moisture)
## [1] 1.924749
```

e)

In order to compare the differences with and without the spatial trend, run a simple regression of the two variables:

```
#parameter estimates with spatial adjustment
sph.fit.nt$beta
##  intercept     covar1
## -0.3044718 -0.5787528
#and their standard errors
sqrt(c(sph.fit.nt$beta.var[1,1],sph.fit.nt$beta.var[2,2]))
## [1] 0.09110294 0.06008915
#a simple linear model with no spatial adjustment
nospat <- lm( formula=moist$logm~moist$density)
summary(nospat)$coefficients
##                 Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)   -0.1863549 0.08909286  -2.091693 3.748336e-02
## moist$density -0.6573625 0.05982847 -10.987452 3.839276e-23
```

It is clear from the output that the estimates and the SEs between the two models will be different when spatial adjustments are accounted for, but both the intercept and the density are still significant and the differences are not too large. (We could also just use the $nospatial from the likfit() output)

4)

```
arse <- read.table("arsenic.txt", header = T)
missing = (arse$depth_m == 0) | (arse$year_made == -1)
arsenic = arse[!missing,]
arsenic$logar <- log(arsenic$arsenic)
as_geo  = as.geodata(obj = arsenic, coords.col = 2:1, data.col = 8)
arvar <-variog(as_geo, uvec = seq(0.1,3.5,length.out = 50))
```

I use my own function for cross-validation instead of geoR's:

```
arsesim <- function(rawd,ntimes){
  options(warn = -1, max.print = 15)
  mtr.mspe = exp.mspe = exp.sill = exp.range = exp.nugget = mtr.sill = mtr.range = mtr.nugget <- matrix()
  modeltype <- c(rep("exp", times = ntimes),rep("matern", times = ntimes))
  number <- c(rep(1:ntimes, times = 2))
  for(i in 1:ntimes){
  set.seed(1234*i^2)
  subsample = sample.int(nrow(rawd), size = length(rawd[,1])/2)
  geot = jitterDupCoords(as.geodata(obj = rawd[-subsample,],coords.col = 2:1 ,data.col = 8),max = .0002)
  geoval = jitterDupCoords(as.geodata(obj = rawd[subsample,],coords.col = 2:1 ,data.col = 8),max = .0002)
  arvar <-variog(geot, uvec = seq(0.1,3.5,length.out = 50))
  pred.locs <- rawd[subsample,2:1]
  ar.exp <- variofit(arvar,ini.cov.pars = c(5,1), cov.model = "exponential", weights = "cressie",
                     nugget = 2)
  ar.mtr <- variofit(arvar,ini.cov.pars = c(5,1), cov.model = "matern", weights = "cressie",
           nugget = 2, kappa = .25, fix.kappa = T)
  pred.exp = krige.conv(geodata=geot, locations= pred.locs,
                                krige=krige.control(type.krige="OK",obj.model = ar.exp))
  pred.mtr = krige.conv(geodata=geot, locations= pred.locs,
                        krige=krige.control(type.krige="OK",obj.model = ar.mtr))
  exp.mspe[i] <- mean((pred.exp$predict-geoval$data)^2)
  mtr.mspe[i] <- mean((pred.mtr$predict-geoval$data)^2)
  exp.sill[i] <- ar.exp$cov.pars[1]
  exp.range[i] <- ar.exp$cov.pars[2]
  mtr.sill[i] <- ar.mtr$cov.pars[1]
  mtr.range[i] <- ar.mtr$cov.pars[2]
  mtr.nugget[i] <- ar.mtr$nugget
  exp.nugget[i] <- ar.exp$nugget
  }
  outlist <- tibble(number,modeltype,c(exp.mspe,mtr.mspe), c(exp.sill,mtr.sill),c(exp.range,mtr.range),
```

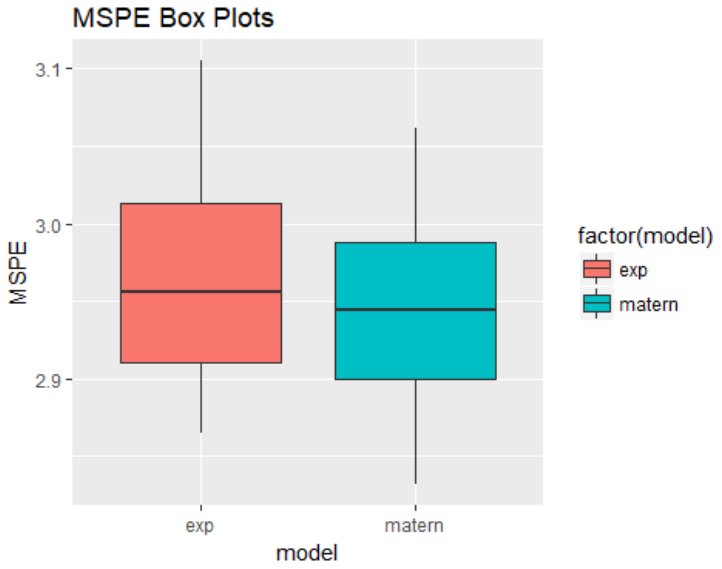
```
c(exp.nugget, mtr.nugget))
  names(outlist) <- c("number","model", "mspe", "sill", "range", "nugget")
  return(outlist)
}
```

And now we can run the simulation 20 times quickly:

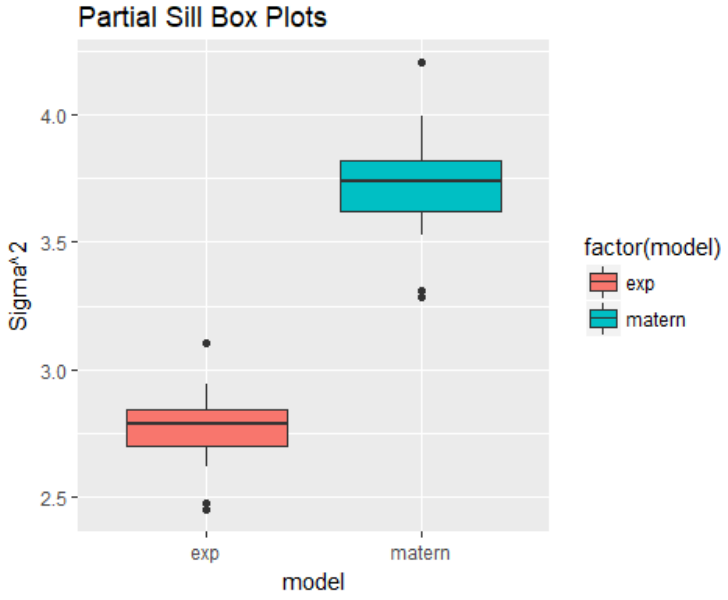
```
sims <- arsesim(arsenic,20)
sims %>% arrange(number) %>% head()
##   number  model     mspe     sill     range   nugget
## 1      1    exp 2.953693 2.688144 0.9039495 3.175556
## 2      1 matern 2.901486 3.610103 1.3908275 2.378883
## 3      2    exp 2.940893 2.830387 0.7262465 3.051621
## 4      2 matern 2.921282 3.761915 1.1117211 2.231548
## 5      3    exp 3.005245 2.720084 0.9941621 3.238871
## 6      3 matern 2.955559 3.665664 1.5908878 2.456667
```

The results as requested. We can see that the MSPE for the simulated matern models is slightly lower and has smaller variance than the exponential models, and that there are large differences in the parameter estimates (as should be expected):

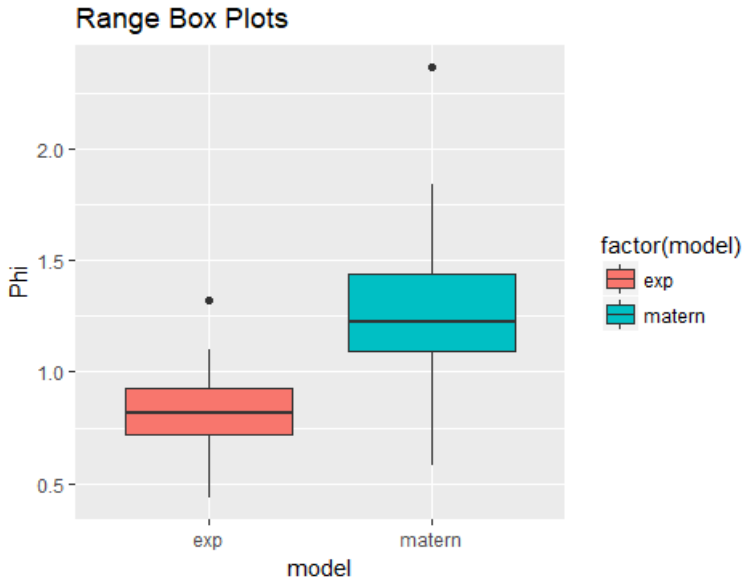
```
bb <- ggplot(data = sims, aes(x = factor(model)))
bb + geom_boxplot(aes(y=mspe, fill = factor(model))) + ylab("MSPE") + xlab("model") + ggtitle("MSPE Box
Plots")
```



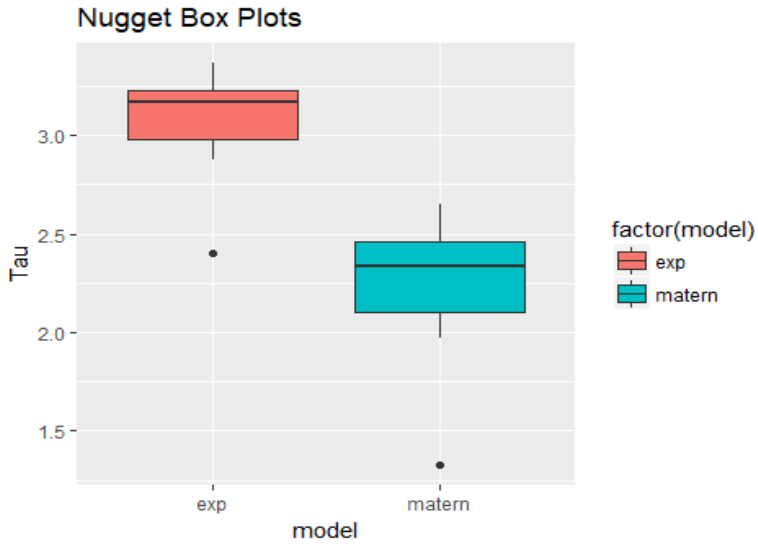

```
bb + geom_boxplot(aes(y=sill, fill = factor(model))) + ylab("Sigma^2") + xlab("model") + ggtitle("Partial Box
Plots")
```



```
bb + geom_boxplot(aes(y=range, fill = factor(model))) + ylab("Phi") + xlab("model") + ggtitle("Range Box Plots")
```




```
bb + geom_boxplot(aes(y=nugget, fill = factor(model))) + ylab("Tau") + xlab("model") + ggtitle("Nugget Box Plots")
```




```
sims %>% group_by(model) %>% summarize(sill.mean = mean(sill), sill.sd = sd(sill), range.mean = mean(range), range.sd = sd(range), nugget.mean = mean(nugget), nugget.sd = sd(nugget))

## # A tibble: 2 x 7
##    model sill.mean   sill.sd range.mean  range.sd nugget.mean nugget.sd

## 1    exp  2.770210 0.1536627  0.8160018 0.2080327    3.092100 0.2200340
## 2 matern  3.723252 0.2133542  1.2663178 0.4058278    2.260749 0.2918012
```