

Content:

Section 1: Import data

Section 2: Calculate digital soil map

Section 3: Clustering

Section 4: Sample size

Section 5: Design effect

Section 6: Sampling design

Section 7: Space-time sampling

```
#Load packages
library(sf)
library(rgdal)
library(raster)
library(sp)
library(tidyverse)
library(Hmisc)
library(factoextra)
library(ggcorrplot)
library(plotly)
library(sampling )
library(surveyplanning)
library(BalancedSampling)
library(pastecs)

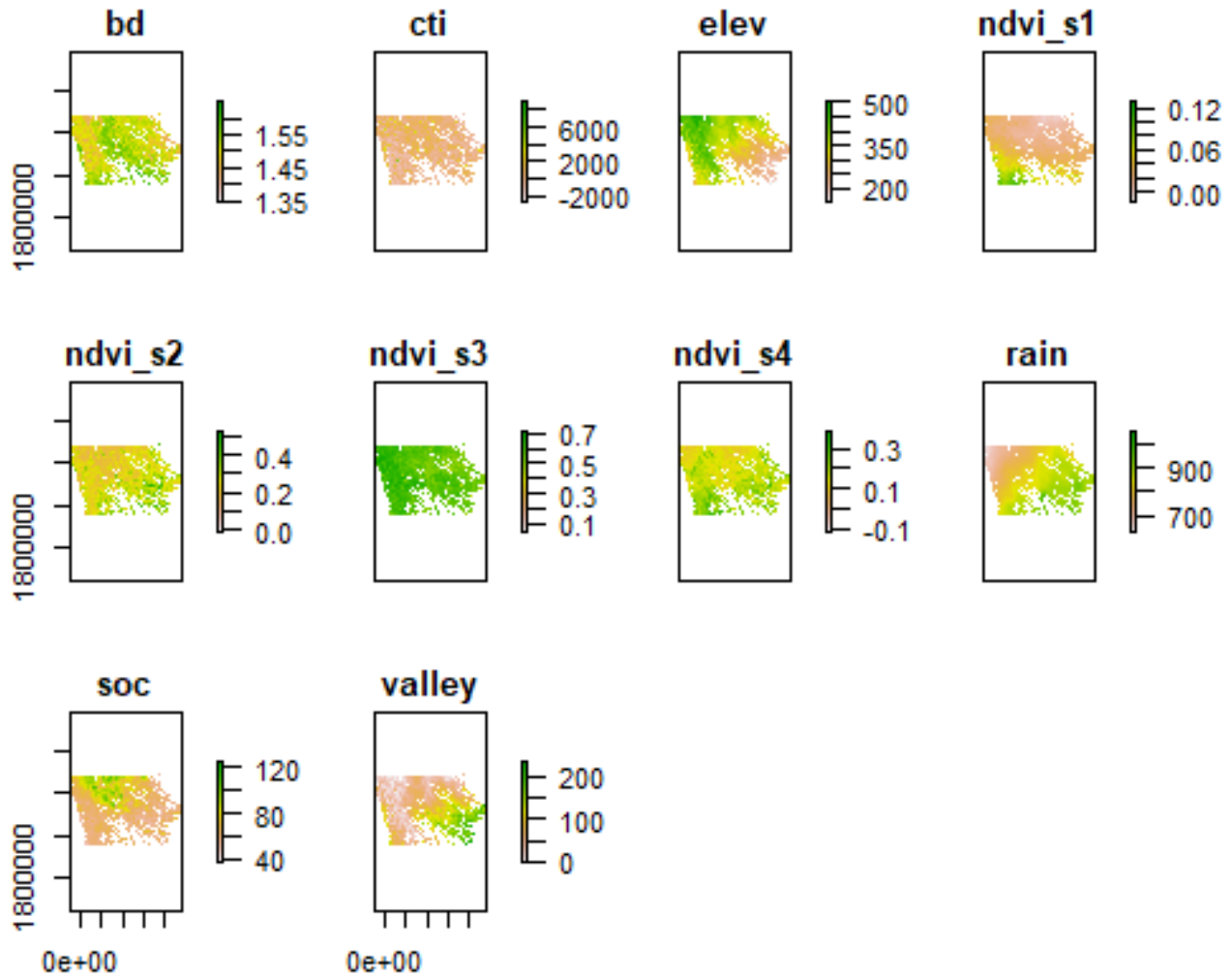
# predefined parameters (figures)
t1 <- theme(axis.text.x = element_text(face = "bold", color = "#993333",size
= 15), axis.text.y = element_text(face = "bold", color = "blue",size = 15))+
  theme(legend.title = element_text(size = 15))+
  theme(axis.title.x = element_text(size=15))+
  theme(axis.title.y = element_text(size=15))
# working directory
setwd("D:/InputData")

# Load shape files
bound <- read_sf("iowa_b.shp")
socs_poi <- read_sf("soil_p.shp")
socs_crop <- read_sf("crop_b.shp")
```

Section 1

load raster files:

```
ras_lst <- list.files("./cov", pattern="\\.tif$", full.names = TRUE)
cov <- stack(ras_lst)
plot(cov, new = TRUE)
```



calculate carbon stocks in soils (ton/ha.year)

```
soil_weight_kg = cov$bd * 0.3 * 10000 * 1000
cov$socs <- ((cov$soc * soil_weight_kg) / (1000 * 1000))
```

convert stack layers into data frame, remove NA values, and inspect data

```
df_cov <- as.data.frame(cov, xy=T)
df_cov <- df_cov[complete.cases(df_cov),]
```

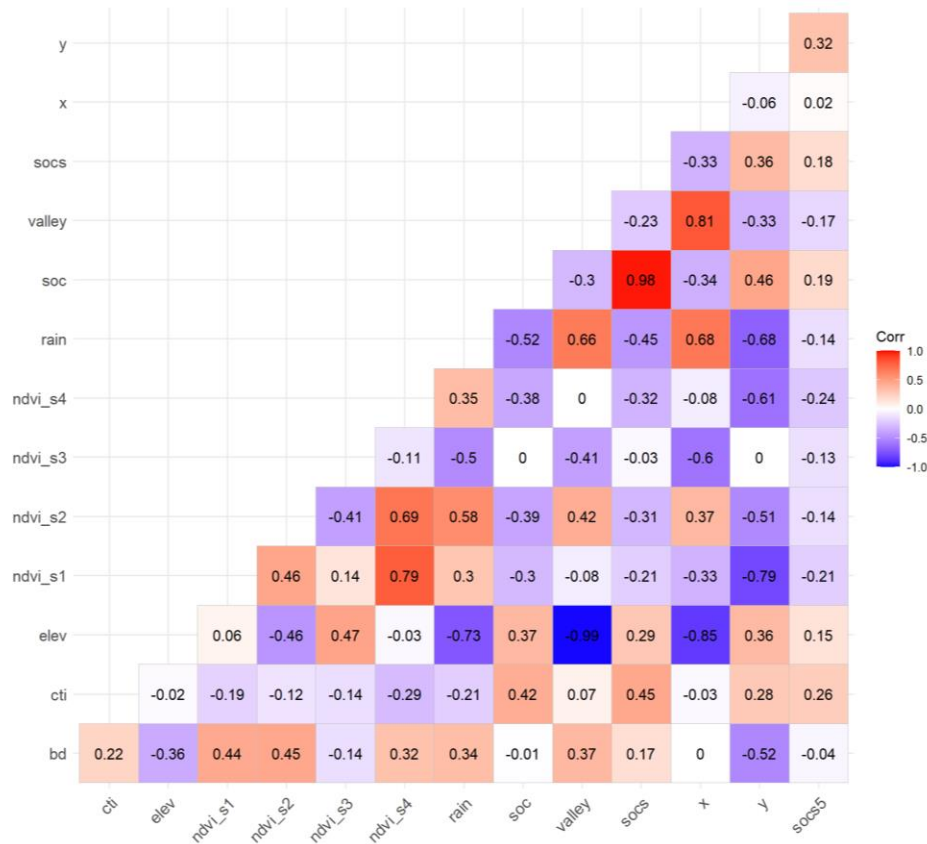
Section 2

extract values at the points

```
cov_socs <- raster::extract(cov,socs_poi, method='bilinear', df=TRUE)
cov_socs <- cov_socs %>% add_column(x= socs_poi$Gen_long,y= socs_poi$Gen_lat,
socs5= socs_poi$SOCstock5)
head(cov_socs)
```

compute a correlation between socs and the other parameters

```
cov_socs_scale <- scale(cov_socs[, -1])
corr <- round(cor(cov_socs_scale), 2)
ggcorrplot(corr, hc.order = F, type = "lower", lab = TRUE)
```



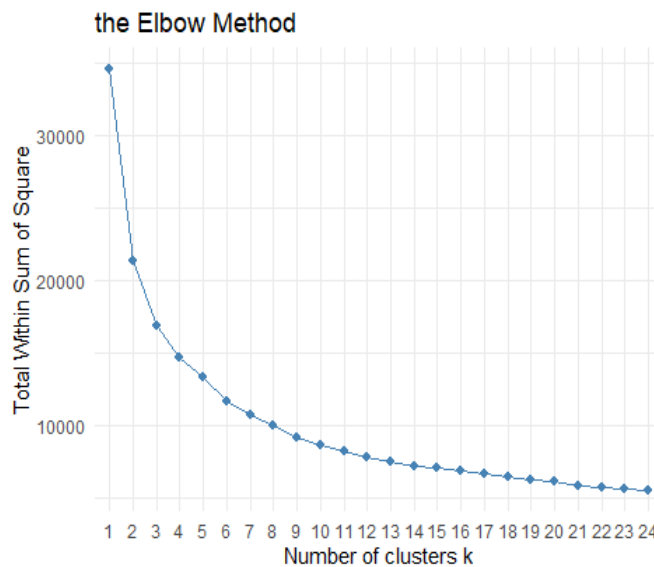
digital soil mapping of socs

```
lm_socs <- lm(socs5 ~ cti + ndvi_s1 + ndvi_s4 + soc, data = cov_socs)
lm_pr_re <- predict(lm_socs, df_cov, se.fit = T)
lm_pr_p_re <- predict(lm_socs, cov_socs, se.fit = T)
df_cov$socs_pre <- lm_pr_re$fit
df_cov$socs_er <- lm_pr_re$se.fit
cov_socs$socs_pre <- lm_pr_p_re$fit
cov_socs$socs_er <- lm_pr_p_re$se.fit
```

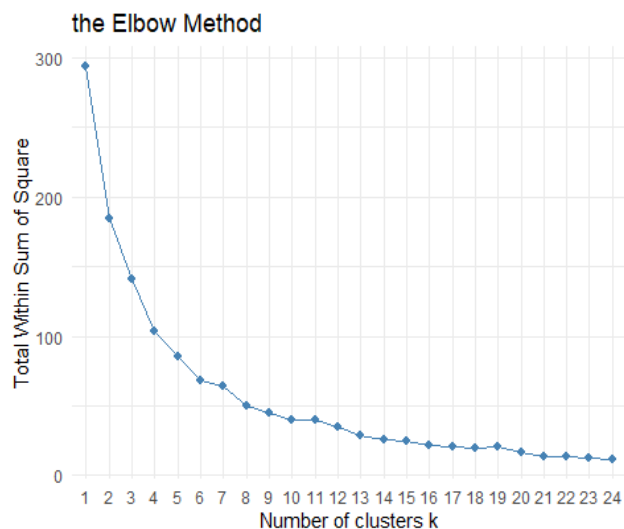
Section 3

```
# clustering section
# prepare data.frame for clustering
df_cov_clus = df_cov %>% select(y,cti,ndvi_s1,ndvi_s4,socs_pre,socs_er) %>% sample_frac(0.05) %>% scale()
df_socs_clus = cov_socs %>% select(y,cti,ndvi_s1,ndvi_s4,socs_pre,socs_er) %>% scale()

# optimize number of clusters
fviz_nbclust(df_cov_clus, kmeans, method = "wss", k.max = 24, iter.max=50) + theme_minimal() + ggtitle("the Elbow Method")
```



```
fviz_nbclust(df_socs_clus, kmeans, method = "wss", k.max = 24, iter.max=50) + theme_minimal() + ggtitle("the Elbow Method")
```

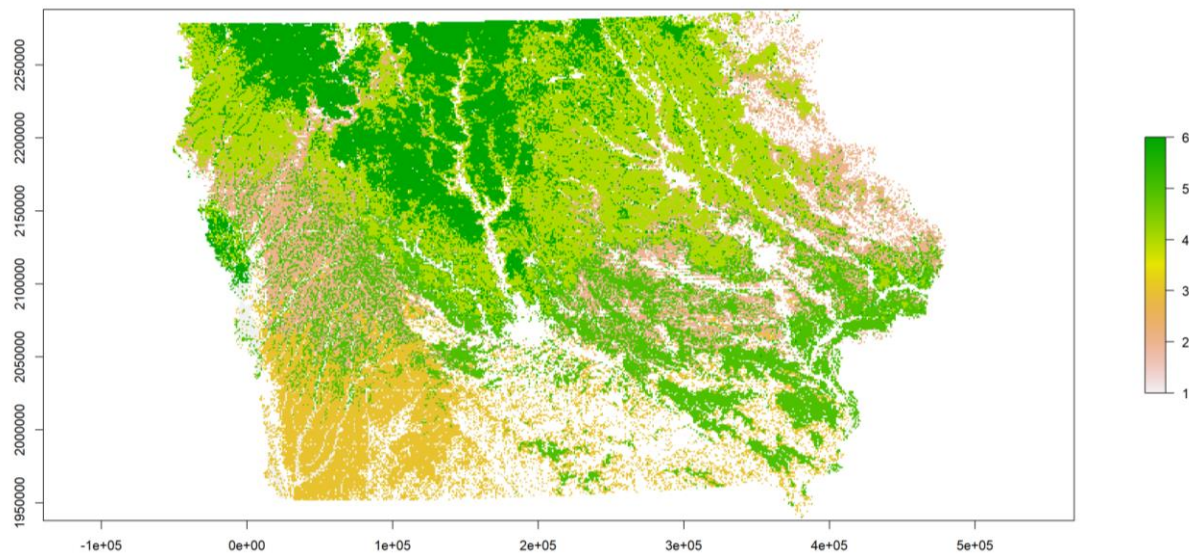


```
# clustering point data
```

```
cls_opt_p <- kmeans(df_socls_clus, centers = 6, nstart = 25, iter.max = 1000)  
cls_p_df <- cov_socls %>% select(x,y,socls5) %>% add_column(cls=as.factor(cls_opt_p$cluster) )
```

```
# clustering all covariates
```

```
df_cov_cls = df_cov %>% select(y,cti,ndvi_s1,ndvi_s4,socls_pre,socls_er) %>% scale()  
cls_opt <- kmeans(df_cov_cls, centers = 6, iter.max = 1000)  
cls_cov_df <- df_cov %>% select(x,y) %>% add_column(cls=cls_opt$cluster)  
cls_cov = rasterFromXYZ(cls_cov_df)  
proj4string(cls_cov) <- crs(socls_crop)  
cov$cls <- resample(cls_cov, cov, method="ngb")  
plot(cov$cls )
```

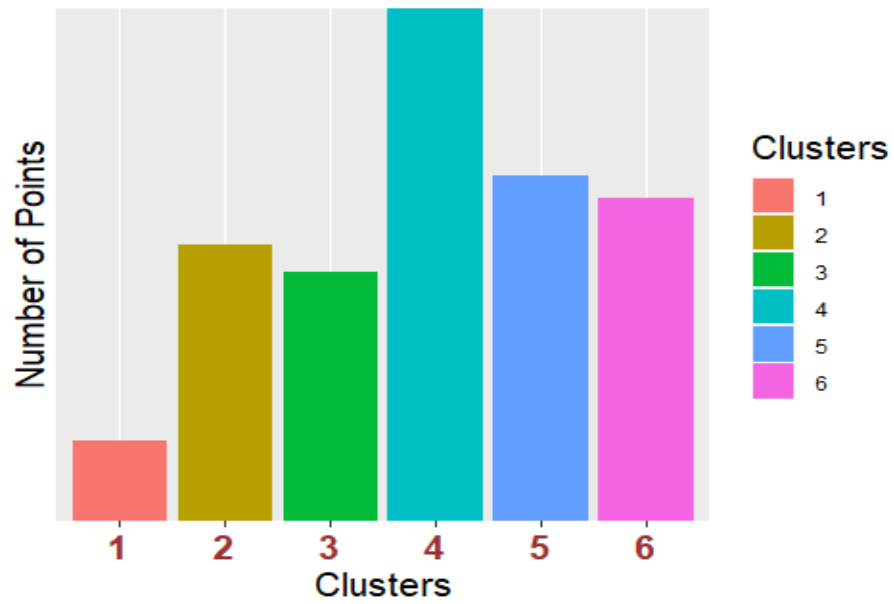


```
# convert to data frame
```

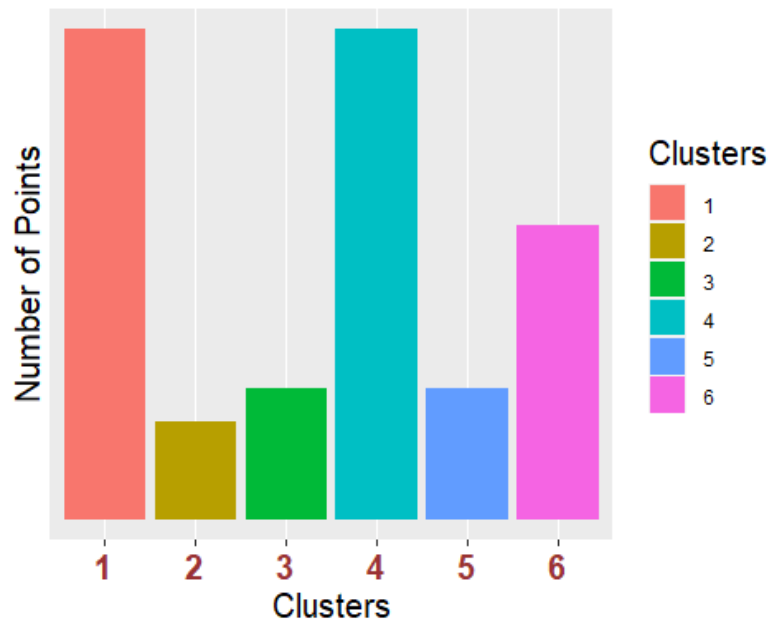
```
df_cov_cls = as.data.frame(cov, xy=T)  
df_cov_cls = df_cov_cls[complete.cases(df_cov_cls),]  
df_cov_cls$cls <- as.factor(df_cov_cls$cls)
```

```
# plot the number of point in each cluster
```

```
ggplot(df_cov_cls) + geom_bar(aes(x = cls, fill = cls)) +t1 +  
  scale_x_discrete(name="Clusters")+  
  scale_y_discrete(name="Number of Points")+  
  labs(fill = "Clusters")
```



```
ggplot(cls_p_df) + geom_bar(aes(x = cls, fill = cls))+t1 +  
  scale_x_discrete(name="Clusters")+  
  scale_y_discrete(name="Number of Points")+  
  labs(fill = "Clusters")
```



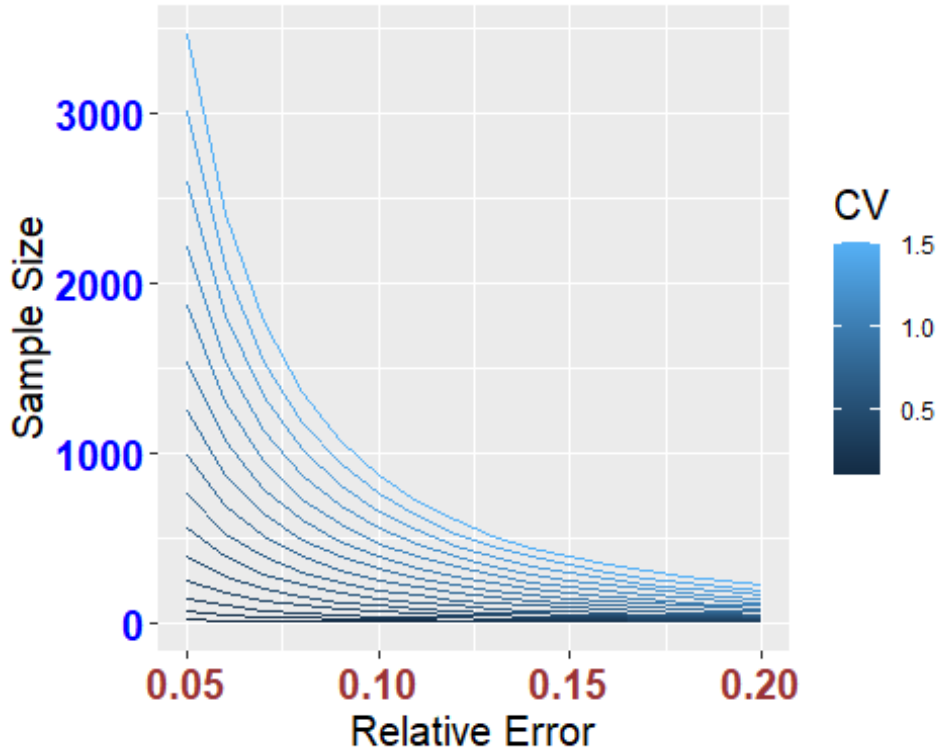
Section 4

```
# calculate the sample size
# cv vs size
cv <- c(seq(0.1, 1.5, by = 0.1) )
rmax <- c(seq(0.05, 0.2, by = 0.01))
u <- qnorm(p=1-0.05/2, mean=0, sd=1)
SI_n = data.frame()

for (i in 1:length(cv)) {
  for (j in 1:length(rmax)) {
    n <- ceiling((u*cv[i]/rmax[j])^2)
    SI_n[i,j] = n }}

names(SI_n) <- c(rmax)
SI_n <- SI_n %>% gather(RE, size, everything()) %>% add_column(CV=rep(cv, 1
length(rmax)))
SI_n$RE <- as.numeric(SI_n$RE )

SI_p <- ggplot(SI_n,aes(x=RE, y=size, group=CV)) + geom_line(aes(color=CV)) +
  scale_x_continuous(name="Relative Error")+
  scale_y_continuous(name="Sample Size")+
  labs(fill = "CV")+t1
SI_p
```



```

# MDD vs size
sd_p <- c(seq(1, 30, by = 1))
size <- c(seq(10, 2000, by = 100))
MDC = data.frame()

for (i in 1:length(sd_p)) {
  for (j in 1:length(size)) {
    n <- ((u*sd_p[i])*((2/size[j])**0.5))
    MDC [i,j] = n }}

breaks = c(seq(0, 4, by=0.1), c(5,6,8,10,12,27))
colnames(MDC) <- c(size)
SI_n_MDC <- MDC %>% gather(size, MDC, everything()) %>%
  add_column(SD=rep(sd_p, length(size)))%>%
  mutate(MDC_C=cut(MDC,breaks=breaks))

SI_n_MDC$size=as.numeric(SI_n_MDC$size)
SI_n_MDC$SD=as.numeric(SI_n_MDC$SD)

SI_n_MDC_p <- ggplot(SI_n_MDC, aes(SD, size, fill=MDC_C)) + geom_tile()+
  geom_text(aes(label = round(MDC, 2))) +
  scale_x_continuous(name="Standard Deviation (T C/ha)", breaks=c(5, 17, 20,
25, 30))+
  scale_y_continuous(name="Sample Size", breaks=c(150, 250, 500, 750,810, 100
0, 1500))+
  geom_rect(data=(data.frame(SD=17,size=810)), size=1, fill=NA, colour="black
",
  aes(xmin=SD - 0.5, xmax=SD + 0.5, ymin=size - 50, ymax=size + 50)
) +
  labs(fill = "MDC (T C/ha.y)")+t1

SI_n_MDC_p

```

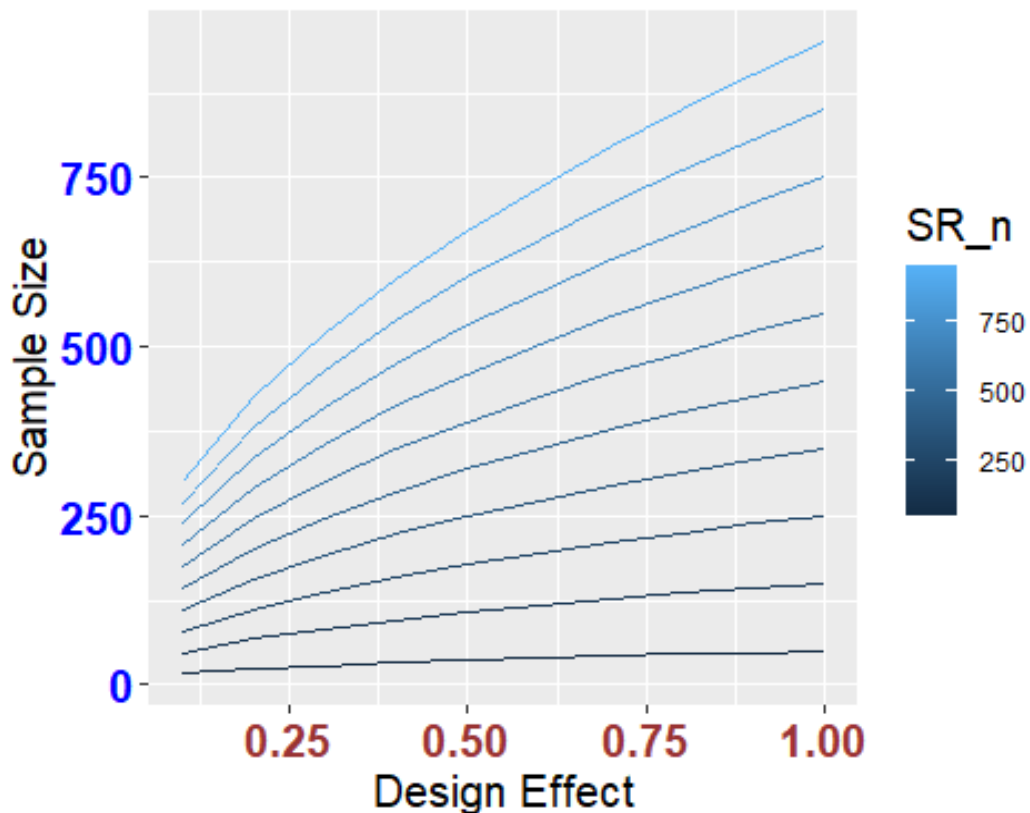

Section 5

```
# design effect
test_n <- c(seq(50,1000,by=100))
de <- c(seq(0.1, 1, by = 0.1))
SIS_n = data.frame()

for (i in 1:length(test_n)) {
  for (j in 1:length(de)) {
    n2 <- test_n[i] * (de[j])**0.5
    SIS_n [i,j] = ceiling(n2) }}

colnames(SIS_n) <- c(de)
SIS_n <- SIS_n %>% gather(design_effect, size, everything()) %>% add_column(
  (SR_n=rep(test_n, length(de))))
SIS_n$design_effect = as.numeric(SIS_n$design_effect )

SIS_n_p <-ggplot(SIS_n,aes(x=design_effect, y=size, group=SR_n)) + geom_line(
  aes(color=SR_n)) +
  scale_x_continuous(name="Design Effect")+
  scale_y_continuous(name="Sample Size")+
  labs(fill = "Simple Random Size")+t1
SIS_n_p
```



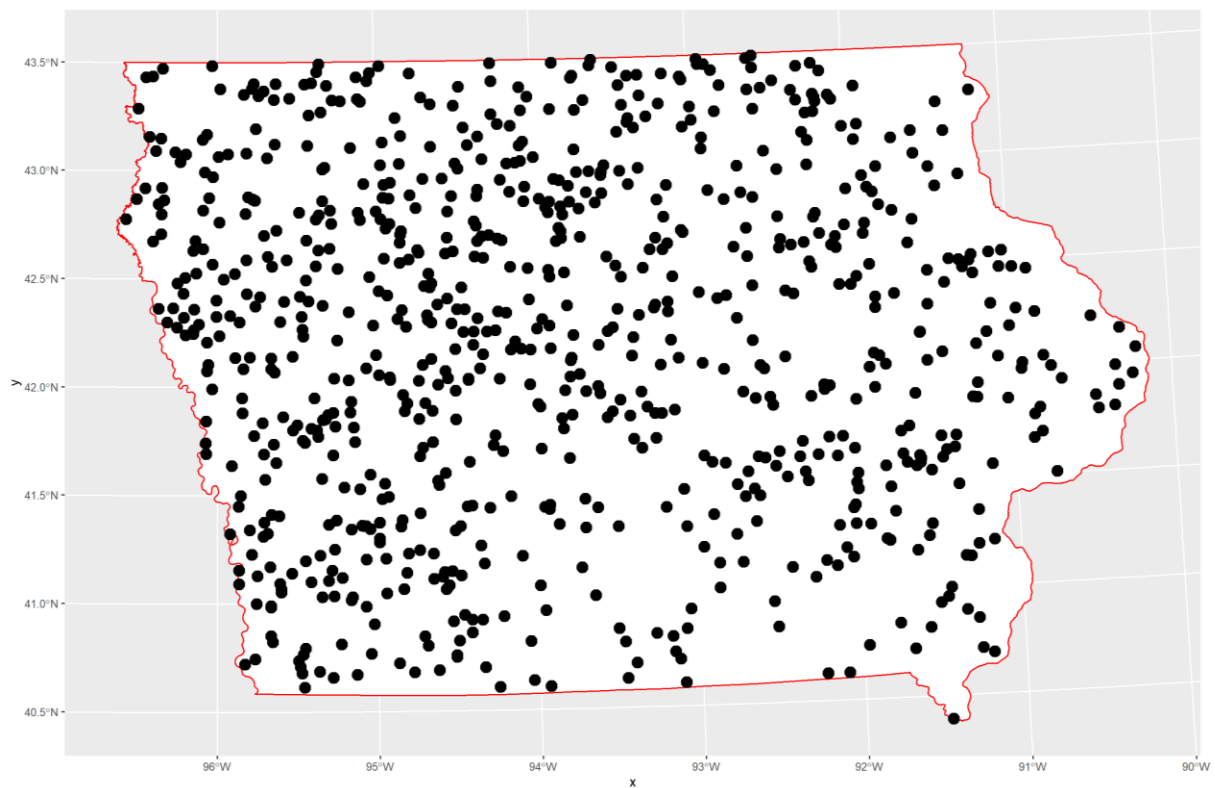
Section 6

```
# sample size for this study
# SR
SR_n_size <- 815
SRS_n_size <- ceiling((0.70)**2 * SR_n_size)
SWB_n_size <- ceiling((0.35)**2 * SR_n_size)
df_cov_s <- df_cov_cls %>% select(x,y,cls)
jit_size <- res(cov)[1] / 2
N <- nrow(df_cov_s)

# spatial distribution
# SI spatial distribution
SI_units <- sample(N, size = SR_n_size, replace = FALSE)
SI_sample <- df_cov_s[SI_units,]
SI_sample$x <- jitter(SI_sample$x, amount = jit_size)
SI_sample$y <- jitter(SI_sample$y, amount = jit_size)

SI_spap <- ggplot(bound) + geom_sf( colour="red",fill="gray100") +
  geom_point(data = SI_sample, aes(x = x, y = y), size = 4,
            shape = 19) + theme_gray()

SI_spap
```



```

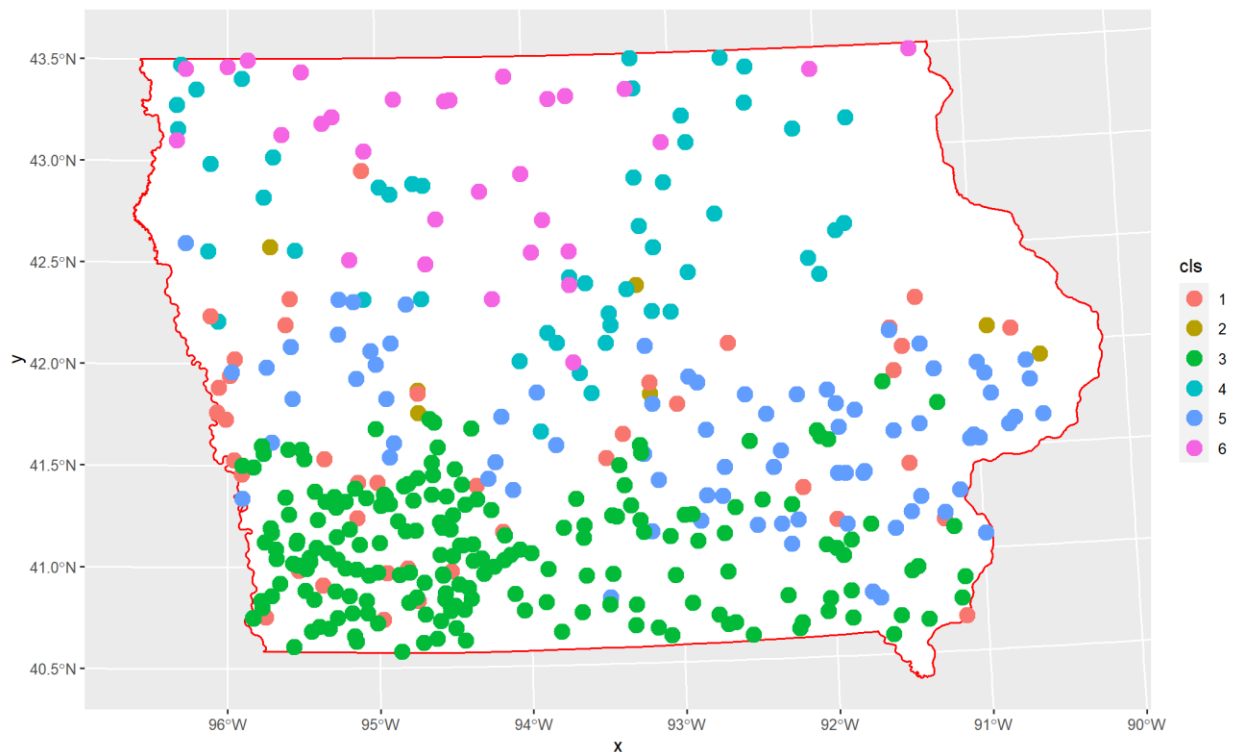
# SRS spatial distribution
N_h <- tapply(df_cov_s$cls, INDEX=df_cov_s$cls, FUN=length)
var_h <- tapply(X=cls_p_df$socs5, INDEX=cls_p_df$cls, FUN=var)
labels <- sort(unique(cls_p_df$cls))
res <- optsize(labels,SRS_n_size,N_h,var_h)
n_size_h <-round(res$nh,0)

SRS_units <- sampling::strata(
  df_cov_s, stratanames="cls", size=n_size_h, method="srswor")
SRS_sample <- getdata(df_cov_s, SRS_units)
SRS_sample$x <- jitter(SRS_sample$x, amount=jit_size)
SRS_sample$y <- jitter(SRS_sample$y, amount=jit_size)

SRS_spa_p <- ggplot(bound) + geom_sf( colour="red",fill="gray100") +
  geom_point(data = SRS_sample, aes(x = x, y = y, color = cls), size = 4,
    shape = 19) + theme_gray()

SRS_spa_p

```



```

# well spread and balanced spatial distribution
bal <- cbind(rep(1,times=N), df_cov$ndvi_s4, df_cov$socs_pre)
spread <- cbind(df_cov$x, df_cov$y)
pi <- rep(SWB_n_size/N, times=N)
SWB_units <- lcube(Xbal=bal, Xspread=spread, prob=pi)
SWB_sample <- df_cov[SWB_units,]

SWB_spa_p <- ggplot(bound) + geom_sf( colour="red",fill="gray100") +
  geom_point(data = SWB_sample, aes(x = x, y = y), size = 4,
            shape = 19) + theme_gray()

SWB_spa_p

```



Section 7

the space-time design

```
STS_df <- data.frame(ID= c(1:nrow(SWB_sample)), Time_01= c(1),Time_02= c(2))
```

```
STS_df_p <- ggplot(STS_df) + geom_point(aes(x=ID, y=Time_01))+  
  geom_point(aes(x=ID, y=Time_02))+  
  scale_x_continuous(name="Space") +  
  scale_y_continuous(name="Time", breaks=c(1, 2))+t1  
STS_df_p
```

