Was the balanced stratified coverage sampling not well spatially distributed, or was it inadequately implemented?

Abdelkrim Bouasria

2025-01-06

## Introduction

This repository contains a reproducible notebook accompanying the comment on [Asrat et al. (2024)](https://doi.org/10.1016/j.geoderma.2024.117116), published in *Geoderma*. The comment is entitled *Was the balanced stratified coverage sampling not well spatially distributed, or was it inadequately implemented?* Please refer to the comment for further details (Bouasria, 2025)

In this notebook, I implement the sampling design described in Asrat et al. (2024) using the details provided in the main paper and the supplementary material. I will demonstrate the preparation of the covariates, followed by the execution of ten sampling iterations to evaluate whether the results exhibit clustering in any iteration.

## Covariates

Using the same covariates and following nearly identical steps, we performed ten sampling iterations, each selecting 599 samples, to evaluate whether the distribution exhibited consistent trends. We focused on the primary wheat-growing regions and applied the cropland mask (GFSAD30AFCE; [Xiong et al., 2017](https://doi.org/10.5067/MEaSUREs/GFSAD/GFSAD30AFCE.001)) at a 30-meter resolution to exclude other land use areas. The FAO soil vector map was rasterized to match the cropland raster grid, and the remaining covariates were aggregated in accordance with the methodology described by Asrat et al. (2024), then resampled to a spatial resolution of 30 meters.

### Definition of wheat-growing areas in Morocco

The soil sampling locations were selected to capture soil variability and distribution across Morocco’s rainfed wheat-growing regions, with a focus on prioritizing soil types relevant to wheat cultivation. We referred to the official statistics of Morocco (CGDA, 2020; page 96) to select 6 out of the 10 wheat-producing regions, which encompass the primary rainfed cropland areas.

The preparation of covariates was performed using the terra package.

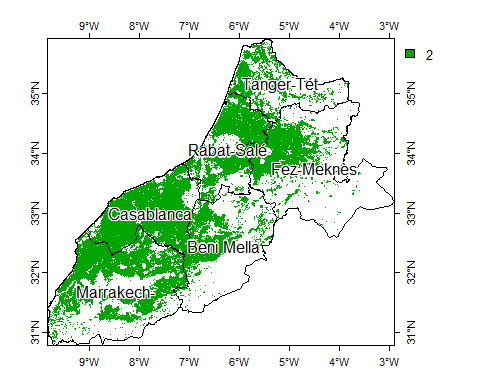
library(terra)

## terra 1.7.65

We sourced cropland data from the GFSAD30AFCE dataset ([Xiong et al., 2017](https://doi.org/10.5067/MEaSUREs/GFSAD/GFSAD30AFCE.001)). Three tiles were combined into a single VRT file, where we assigned NA values to non-cropland classes. We then cropped this file to align with the wheat-growing regions mask.

cropland = rast("./\_RAW\_Data/CROPLAND/croplands.vrt")  
cropland[cropland==0] = NA  
cropland[cropland==1] = NA  
  
rv= vect("./vect/Morocco\_regions\_mask.shp")  
  
crop(cropland,rv, mask=TRUE, filename="./covariates/croplands.tif",  
 datatype="INT1U", gdal=c("COMPRESS=DEFLATE"))

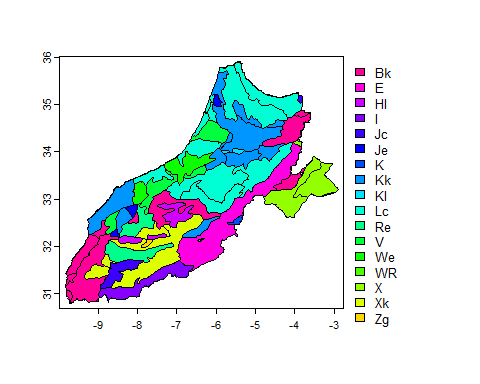
rv= vect("./vect/Morocco\_regions.shp")  
cropland = rast("./covariates/croplands.tif")  
plot(cropland, pax=list(retro=TRUE, side=c(1:4)))  
polys(rv)  
text(rv,"NAME\_1", halo=TRUE)



### FAO Soil map

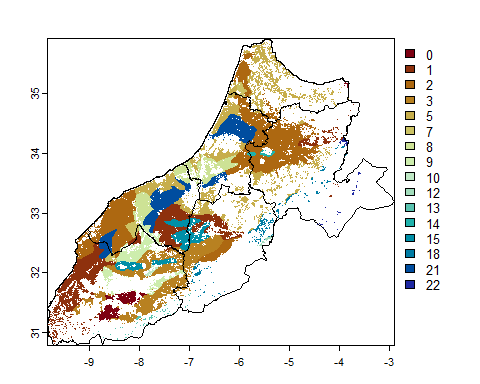
Soil data was obtained

rv\_mask= vect("./vect/Morocco\_regions\_mask.shp")  
soil = crop(vect("./\_RAW\_Data/FAO\_SOIL/DSMW/DSMW.shp"),rv\_mask)  
plot(soil,"DOMSOI", ext=ext(rv))



mask(rasterize(soil,cropland, field="Legend"),  
 cropland, filename="./covariates/fao\_soil.tif",  
 datatype="INT1U", gdal=c("COMPRESS=DEFLATE"))

rst\_soil = rast("./covariates/fao\_soil.tif")  
plot(rst\_soil, type="classes", col=hcl.colors(23,"Roma"))  
polys(rv)



### Topographic covariates: elevation and slope

DEM data was sourced from xxxx using QGIS plugin

library("marmap")

## Warning: package 'marmap' was built under R version 4.2.3

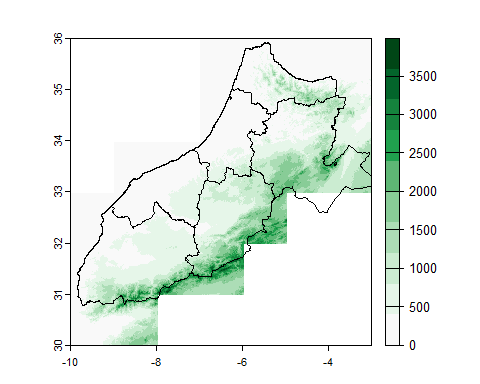
## Registered S3 methods overwritten by 'adehabitatMA':  
## method from  
## print.SpatialPixelsDataFrame sp   
## print.SpatialPixels sp

##   
## Attaching package: 'marmap'

## The following object is masked from 'package:terra':  
##   
## as.raster

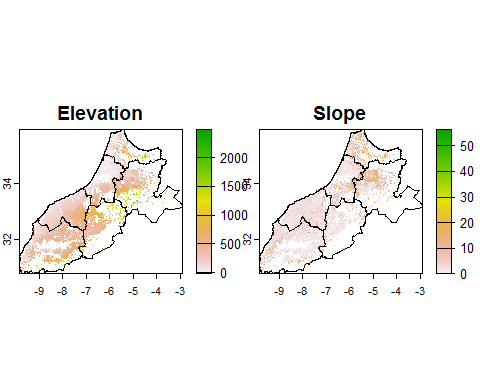
## The following object is masked from 'package:grDevices':  
##   
## as.raster

dem = rast("./\_RAW\_Data/DEM/dem.vrt")  
plot(dem, col= rev(hcl.colors(10, "Greens 3")))  
polys(rv)

 The elevation and slope was prepared using

resample(crop(dem,rv\_mask, mask=TRUE),  
 cropland, method="bilinear", threads=TRUE,  
 filename="./covariates/elevation.tif",  
 gdal=c("COMPRESS=DEFLATE"),overwrite=TRUE)  
  
slope = resample(crop(terrain(dem, v="slope"),rv\_mask, mask=TRUE),  
 cropland, method="bilinear", threads=TRUE,  
 filename="./covariates/slope.tif",  
 gdal=c("COMPRESS=DEFLATE"),overwrite=TRUE)

elevation = rast("./covariates/elevation.tif")  
slope = rast("./covariates/slope.tif")  
  
plot(c(elevation,slope), main=c("Elevation","Slope"),  
 fun = \() polys(rv))



### Rainfall: Five years annual average

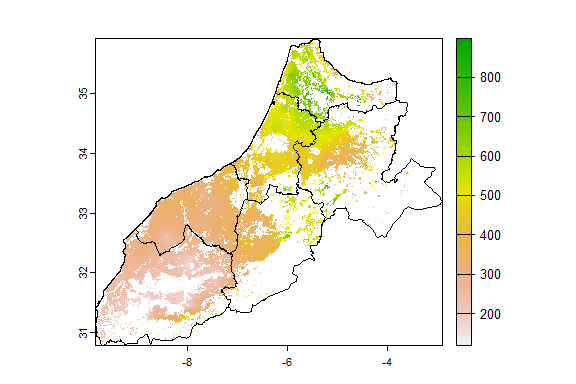
We decompressed and cropped the CHIRPS decadal data of Africa to the study area.

library(R.utils)  
# Crop for the study area extent  
dir\_in = "./\_RAW\_Data/CHIRPS/"  
dir\_out = "./\_RAW\_Data/temporary/"  
gz\_files = list.files(dir\_in, pattern = ".gz$")  
  
  
for(gz in gz\_files){  
 tif\_files = rast(gunzip(paste0(dir\_in,gz)))  
 terra::crop(tif\_files,rv\_mask, mask=TRUE, filename=paste0(dir\_out,gsub(".gz","",gz)))  
 cat(gz,"\n")  
}

We calculated

# calculate the cumulative annual rainfall  
rst\_files = list.files(dir\_out, pattern = ".tif$")  
years = sapply(gsub(".tif$","",gsub("chirps-v2.0.","",rst\_files)),  
 \(x) strsplit(x,"\\.")[[1]][1])  
  
df\_files = data.frame(files=rst\_files,years=years )  
  
annual\_rainfall = NULL  
  
for(year in unique(df\_files$years)){  
 df = df\_files[df\_files$years==year,]  
 rainfall\_year = sum(rast(paste0(dir\_out,df$files)))  
   
 if(is.null(annual\_rainfall)){  
 annual\_rainfall = rainfall\_year  
 }else{  
 annual\_rainfall = c(annual\_rainfall, rainfall\_year)  
 }  
}  
  
# calculate the five years annual avg  
annual\_rainfall\_avg = mean(annual\_rainfall)  
  
# resemple to cropland raster  
resample(annual\_rainfall\_avg,  
 cropland, method="bilinear", threads=TRUE,  
 filename="./covariates/rainfall.tif",  
 gdal=c("COMPRESS=DEFLATE"),overwrite=TRUE)

rainfall = rast("./covariates/rainfall.tif")  
plot(rainfall)  
polys(rv)



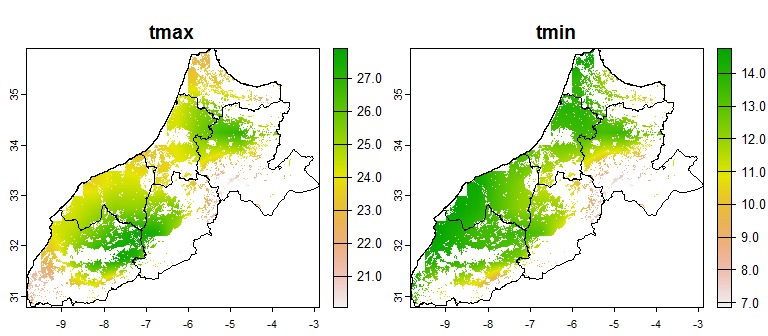
#

### Temperature: Five years annual minimum and maximum temperature

<https://downloads.psl.noaa.gov/Datasets/cpc_global_temp/>

rst\_files = list.files("./Temp\_NOAA/", pattern = ".nc$")  
temp1 = rast(paste0("./Temp\_NOAA/",rst\_files[1]))  
plot(temp1[[1]])  
x <- rotate(temp1)   
plot(x[[1]])  
  
rv1 = buffer(rv,5000)  
  
x1 = crop(x,rv1, mask=TRUE)  
plot(x1[[1]])  
polys(rv)  
polys(rv1)  
  
for(rst in rst\_files){  
 terra::crop(rotate(rast(paste0("./Temp\_NOAA/",rst))),rv1, mask=TRUE, filename=paste0("./Temp\_NOAA\_MA/",rst))  
 cat(rst,"\n")  
}  
  
# calculate annual AVG of Tmin and Tmax  
rst\_files = list.files("./Temp\_NOAA\_MA/", pattern = ".nc$")  
  
years = sapply(gsub(".nc$","",rst\_files),  
 \(x) strsplit(x,"\\.")[[1]][2])  
temp = sapply(gsub(".nc$","",rst\_files),  
 \(x) strsplit(x,"\\.")[[1]][1])  
  
df\_files = data.frame(files=rst\_files,temp=temp,years=years )  
  
df\_tmax = df\_files[df\_files$temp=="tmax",]  
df\_tmin = df\_files[df\_files$temp=="tmin",]  
  
annual\_tmax = lapply(df\_tmax$years, function(year){  
 mean(rast(paste0("./Temp\_NOAA\_MA/",df\_tmax[df\_tmax$years==year,]$files)))  
})  
  
annual\_tmax\_avg = mean(rast(annual\_tmax))  
  
annual\_tmin = lapply(df\_tmin$years, function(year){  
 mean(rast(paste0("./Temp\_NOAA\_MA/",df\_tmin[df\_tmax$years==year,]$files)))  
})  
  
annual\_tmin\_avg = mean(rast(annual\_tmin))  
  
  
tmax = resample(annual\_tmax\_avg,  
 cropland, method="bilinear", threads=TRUE,  
 filename ="./covariates/tmax.tif",   
 gdal=c("COMPRESS=DEFLATE"),overwrite=TRUE)  
tmin = resample(annual\_tmin\_avg,  
 cropland, method="bilinear", threads=TRUE,  
 filename ="./covariates/tmin.tif",   
 gdal=c("COMPRESS=DEFLATE"),overwrite=TRUE)

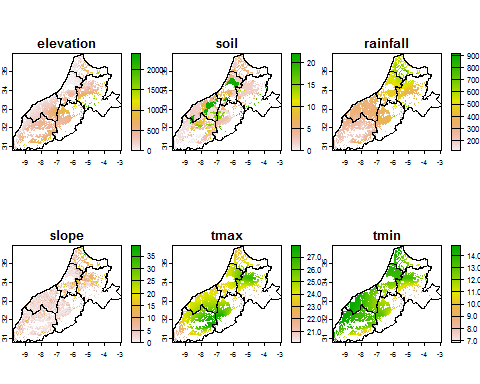
temp = c(rast("./covariates/tmax.tif"),rast("./covariates/tmin.tif"))  
plot(temp, fun= \() polys(rv))



## Sampling design test

The authors utilized the lcubestratified function from the BalancedSampling R package, which implements stratified doubly balanced sampling with pooling of landing phases, utilizing the fast flight Cube method. This function requires seven input arguments, but only four are specified in the paper. These include: (1) the inclusion probabilities (prob), which were set to be equal for all points; (2) the spreading parameter (Xpread), representing the geographical coordinates across the study area; (3) the balancing auxiliary variables (Xbal), which include rainfall, temperature, slope, and elevation; and (4) the stratification parameter (strata), representing the FAO soil types . The settings for the remaining arguments were not reported, and therefore, it is assumed that they were left at their default values.

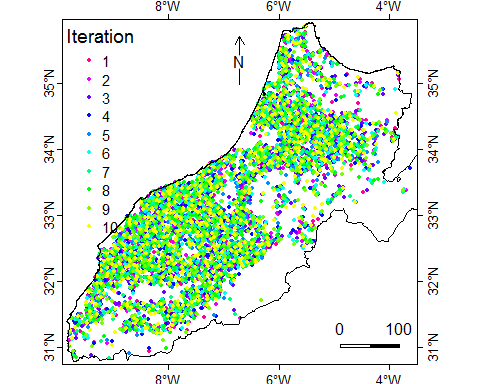
r\_files = list.files("./covariates/", pattern = ".tif$", full.names = T)[-1]  
cov = rast(lapply(r\_files, \(r) rast(r)))  
plot(cov, fun=function() polys(rv))



library(BalancedSampling)  
  
rv= vect("./vect/Morocco\_regions\_mask.shp")  
cov = as.data.frame(cov, xy=T,na.rm=TRUE)  
  
saveRDS(cov,"./covariates/cov\_30m\_df.rds")  
  
N = nrow(cov)  
n = 599  
sprob = rep(n/N, N)  
  
xspr = cov[,c("x","y")]  
xbal = cov[,c("elevation", "slope", "rainfall", "tmax", "tmin")]  
strata = cov[,"soil"]  
  
iter = 10  
set.seed(1)  
seeds <- runif(iter,1,100000)|> round(0)  
  
samples = list()  
  
for(i in 1:iter){  
 set.seed(seeds[i])  
 samples[[i]] = lcubestratified(prob = sprob,  
 Xspread = xspr,  
 Xbal = xbal,  
 integerStrata = strata)  
 cat("iter:",i,"\n")  
}  
  
  
saveRDS(samples,"./samples/samples.rds")  
saveRDS(xspr,"./samples/cov\_xy.rds")

samples = readRDS("./samples/samples.rds")  
xspr = readRDS("./samples/cov\_xy.rds")  
crs\_xy = crs(rast("./cov/cov\_30m.tif"))  
  
plot(vect(xspr[samples[[1]],], geom=c("x", "y"), crs=crs\_xy))  
polys(rv)  
  
samples\_xy = lapply(samples,\(x) xspr[x,])  
for(i in 1:length(samples\_xy)){  
 samples\_xy[[i]]=cbind(samples\_xy[[i]], iter = i)  
}  
samples\_c = do.call(rbind,samples\_xy)  
  
saveRDS(samples\_c,"./samples\_c.rds")  
  
  
samples\_vect = vect(samples\_c, geom=c("x", "y"), crs=crs\_xy)  
writeVector(samples\_vect,"./samples/samples\_vect.gpkg")

rv= vect("./vect/Morocco\_regions\_mask.shp")  
samples\_vect = vect("./samples/samples\_vect.gpkg")  
  
plot(samples\_vect,"iter", cex=0.5,mar=c(1,1,1,1),  
 plg=list(x="topleft",title="Iteration", title.cex=1.2, cex=1.2),   
 pax=list(side=1:4, retro=TRUE,cex.axis=1.2))  
polys(rv)  
north("top")  
sbar(xy="bottomright", type="bar")



par(mfrow = c(3,3))  
  
for(i in 1:9){  
 plot(samples\_vect[samples\_vect$iter==i],cex=0.5, col="blue",ext=ext(rv),  
 main=paste0("sampling iteration: ",i), mar=c(1,1,1.8,0.5))  
 polys(rv)  
}

