

figure.5

Tang Jiaxin

2025-05-21

```
rm(list= ls())
require(terra)

##      terra
## terra 1.8.42
require(data.table)

##      data.table
##
##      'data.table'
## The following object is masked from 'package:terra':
##
##      shift
require(metafor)

##      metafor
##      Matrix
##      metadat
##      numDeriv
##
## Loading the 'metafor' package (version 4.8-0). For an
## introduction to the package please type: help(metafor)

# what rasters are in data
rfiles <- list.files('D:/coursework/tang/ /data', pattern = 'tif$',full.names = TRUE)
rfiles <- rfiles[!grepl('cropland',rfiles)]

# read in raster files
r.ma <- terra::sds(rfiles)

# convert to raster
r.ma <- terra::rast(r.ma)

# convert rasters to data.table

# set first to xy data.frame (NA=FALSE otherwise gridcells are removed)
r.df <- as.data.frame(r.ma,xy = TRUE, na.rm = FALSE)

# convert to data.table
```

```

r.dt <- as.data.table(r.df)

# setnames
setnames(r.dt,old = c('climate_mat', 'climate_pre','soil_isric_phw_mean_0_5','soil_isric_clay_mean_0_5',
                      'nifert_nfert_nh4','nifert_nfert_no3','nofert_nofert','cropintensity_cropintensity'),
        new = c('mat','pre','phw','clay','soc','nh4','no3','nam','cropintensity'),skip_absent = T)

# select only land area
r.dt <- r.dt[!(is.na(mat)|is.na(pre))]
r.dt <- r.dt[!(is.na(tillage_RICE) & is.na(tillage_MAIZ) & is.na(tillage_other) & is.na(tillage_wheat))]
r.dt <- r.dt[!(is.na(ma_crops_RICE) & is.na(ma_crops_MAIZ) & is.na(ma_crops_other) & is.na(ma_crops_wheat))]

# replace area with 0 when missing
cols <- colnames(r.dt)[grepl('^ma_|nh4|no3|nam',colnames(r.dt))]
r.dt[,c(cols) := lapply(.SD,function(x) fifelse(is.na(x),0,x)),.SDcols = cols]
cols <- colnames(r.dt)[grepl('^tillage',colnames(r.dt))]
r.dt[,c(cols) := lapply(.SD,function(x) fifelse(is.na(x),1,x)),.SDcols = cols]
r.dt[is.na(cropintensity), cropintensity := 1]

# melt the data.table
r.dt.melt <- melt(r.dt,
                 id.vars = c('x','y','mat', 'pre','phw','clay','nh4','no3','nam','soc','cropintensity'),
                 measure=patterns(area="^ma_crops", tillage = "^tillage_"),
                 variable.factor = FALSE,
                 variable.name = 'croptype')

# set the crop names (be aware, its the order in ma_crops)
r.dt.melt[,croptype := c('rice','maize','other','wheat')[as.numeric(croptype)]]

# set names to tillage practices
r.dt.melt[, till_name := 'conventional']
r.dt.melt[tillage %in% c(3,4,7), till_name := 'no-till']

# derive the meta-analytical model

# read data
d1 <- readxl::read_xlsx('D:/coursework/tang/ /Source Data.xlsx',sheet = "Figure5")
d1 <- as.data.table(d1)

# add CV for NUE treatment and estimate the SD for missing ones
d2<-d1
CV_nuet_bar<-mean(d2$nuet_sd[is.na(d2$nuet_sd)==FALSE]/d2$nuet_mean[is.na(d2$nuet_sd)==FALSE])
d2$nuet_sd[is.na(d2$nuet_sd)==TRUE]<-d2$nuet_mean[is.na(d2$nuet_sd)==TRUE]*1.25*CV_nuet_bar

CV_nuec_bar<-mean(d2$nuec_sd[is.na(d2$nuec_sd)==FALSE]/d2$nuec_mean[is.na(d2$nuec_sd)==FALSE])
d2$nuec_sd[is.na(d2$nuec_sd)==TRUE]<-d2$nuec_mean[is.na(d2$nuec_sd)==TRUE]*1.25*CV_nuec_bar

# clean up column names
setnames(d2,gsub('\\/', '_'),gsub(' |\\(|\\|\\|', '', colnames(d2)))
setnames(d2,tolower(colnames(d2)))

```

```

# calculate effect size (NUE)
es21 <- escalc(measure = "MD", data = d2,
              m1i = nuet_mean, sd1i = nuet_sd, n1i = replication,
              m2i = nuec_mean, sd2i = nuec_sd, n2i = replication )

# convert to data.tables
d02 <- as.data.table(es21)

# what are the treatments to be assessed
d02.treat <- data.table(treatment = c('ALL',unique(d02$management)))

# what are labels
d02.treat[treatment=='ALL',desc := 'All']
d02.treat[treatment=='EE',desc := 'Enhanced Efficiency']
d02.treat[treatment=='CF',desc := 'Combined fertilizer']
d02.treat[treatment=='RES',desc := 'Residue retention']
d02.treat[treatment=='RFP',desc := 'Fertilizer placement']
d02.treat[treatment=='RFR',desc := 'Fertilizer rate']
d02.treat[treatment=='ROT',desc := 'Crop rotation']
d02.treat[treatment=='RFT',desc := 'Fertilizer timing']
d02.treat[treatment=='OF',desc := 'Organic fertilizer']
d02.treat[treatment=='RT',desc := 'Reduced tillage']
d02.treat[treatment=='NT',desc := 'No tillage']
d02.treat[treatment=='CC',desc := 'Crop cover']

# update the missing values for n_dose and p2o5_dose (as example)
d02[is.na(n_dose), n_dose := median(d02$n_dose,na.rm=TRUE)]

# scale the variables to unit variance
d02[,clay_scaled := scale(clay)]
d02[,soc_scaled := scale(soc)]
d02[,ph_scaled := scale(ph)]
d02[,mat_scaled := scale(mat)]
d02[,map_scaled := scale(map)]
d02[,n_dose_scaled := scale(n_dose)]

# update the database (it looks like typos)
d02[g_crop_type=='marize', g_crop_type := 'maize']

#Combining different factors

d02[tillage=='reduced', tillage := 'no-till']

# # Combining different factors

d02[,fertilizer_type := factor(fertilizer_type,
                             levels = c('mineral','organic', 'combined','enhanced'))]
d02[,fertilizer_strategy := factor(fertilizer_strategy,
                                   levels = c("conventional", "placement","rate","timing"))]
d02[,g_crop_type := factor(g_crop_type,
                           levels = c('maize','wheat','rice'))]

```

```

d02[,rfp := fifelse(fertilizer_strategy=='placement','yes','no')]
d02[,rft := fifelse(fertilizer_strategy=='timing','yes','no')]
d02[,rfr := fifelse(fertilizer_strategy=='rate','yes','no')]
d02[,ctm := fifelse(g_crop_type=='maize','yes','no')]
d02[,ctw := fifelse(g_crop_type=='wheat','yes','no')]
d02[,ctr := fifelse(g_crop_type=='rice','yes','no')]
#d02[,cto := fifelse(g_crop_type=='other','yes','no')]
d02[,ndose2 := scale(n_dose^2)]

# make metafor model

m1 <- rma.mv(yi,vi,
             mods = ~fertilizer_type + rfp + rft + rfr + crop_residue + tillage +
                   cover_crop_and_crop_rotation + n_dose_scaled + clay_scaled + ph_scaled + map_scaled +
                   n_dose_scaled:soc_scaled + ctm:rfp + ctm + ctw + ctr + ctm:mat_scaled + ndose2,
             data = d02,
             random = list(~ 1|studyid), method="REML",sparse = TRUE)

## Warning: Redundant predictors dropped from the model.

# see model structure that need to be filled in to predict NUE as function of the system properties
p1 <- predict(m1,addx=T)

# this is the order of input variables needed for model predictions (=newmods in predict function)
m1.cols <- colnames(p1$X)

# make prediction dataset for situation that soil is fertilized by both organic and inorganic fertilizers
dt.new <- copy(r.dt.melt)

# add the columns required for the ma model, baseline scenario
# baseline is here defined as "strategy conventional", and mineral fertilizers, no biochar, no crop residue
dt.new[, fertilizer_typeenhanced := 0]
dt.new[, fertilizer_type mineral := 1]
dt.new[, fertilizer_type organic := 0]
dt.new[, fertilizer_type combined := 0]
dt.new[, rfp yes := 0]
dt.new[, rft yes := 0]
dt.new[, rfr yes := 0]
dt.new[, crop_residue yes := 0]
dt.new[, cover_crop_and_crop_rotation yes := 0]
dt.new[, cover_crop_and_crop_rotation yes := fifelse(crop_intensity>1,1,0)]
dt.new[, `tillage no-till` := fifelse(till_name == 'no-till',1,0)]
#dt.new[, `tillage no-till` := 0]
dt.new[, ctryes := fifelse(cropname=='rice',1,0)]
dt.new[, ctwyes := fifelse(cropname=='wheat',1,0)]
dt.new[, ctmyes := fifelse(cropname=='maize',1,0)]
dt.new[, ph_scaled := (phw * 0.1 - mean(d02$ph)) / sd(d02$ph)]
dt.new[, clay_scaled := (clay * 0.1 - mean(d02$clay)) / sd(d02$clay)]
dt.new[, soc_scaled := (soc * 0.1 - mean(d02$soc)) / sd(d02$soc)]
dt.new[, n_dose_scaled := scale(nh4+no3+nam)]
dt.new[, ndose2 := scale((nh4+no3+nam)^2)]
dt.new[, map_scaled := (pre - mean(d02$map)) / sd(d02$map)]
dt.new[, mat_scaled := (mat - mean(d02$mat)) / sd(d02$mat)]

```

```

dt.new[, `n_dose_scaled:soc_scaled` := n_dose_scaled*soc_scaled]
dt.new[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.new[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.new[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))

# add predictions to the data.table
cols <- c('pMDmean','pMDse','pMDcil','pMDciu','pMDpil','pMDpiu')
dt.new[,c(cols) := dt.pred]

#####
##### scenario 1 (Nutrient management) #####
# scenario 1. the combination of measures with change in RFR, RFT and BC. (Optimized fertilizer str

# make local copy
dt.s1 <- copy(dt.new)

# baseline mean and sd for total N input
dt.fert.bs <- dt.new[,list(mean = mean(nh4+no3+nam), sd = sd(nh4+no3+nam))]

# update actions taken for scenario 1
dt.s1[, fertilizer_typeenhanced := 1]
dt.s1[, fertilizer_typemineral := 0]
dt.s1[, fertilizer_typeorganic := 1]
dt.s1[, fertilizer_typecombined := 1]
dt.s1[, rfpyes := 1]
dt.s1[, rftyes := 1]
dt.s1[, rfryes := 1]
dt.s1[, crop_residueyes := 0]
dt.s1[, cover_crop_and_crop_rotationyes := 0]
dt.s1[, tillageno_till := 0]
dt.s1[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]
dt.s1[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 )*soc_scaled]
dt.s1[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.s1[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s1[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred.s1 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s1[,c(cols) := dt.pred.s1]

# compare baseline with scenario

# select relevant columns of the baseline
dt.fin <- dt.new[,.(x,y,base = pMDmean,cropname,area)]

```

```

# select relevant columns of scenario 1 and merge
dt.fin <- merge(dt.fin,dt.s1[,.(x,y,s1 = pMDmean,cropname)],by=c('x','y','cropname'))

# estimate relative improvement via senario 1
dt.fin[, improvement := s1 - base]

# estimate area weighted mean relative improvement
dt.fin <- dt.fin[,list(improvement = weighted.mean(improvement,w = area)),by = c('x','y')]

# make spatial raster of the estimated improvement

# convert to spatial raster
r.fin <- terra::rast(dt.fin,type='xyz')
terra::crs(r.fin) <- 'epsg:4326'

# write as output
terra::writeRaster(r.fin,'D:/coursework/tang/ /tif/scenario_1.tif', overwrite = TRUE)

##### scenario 2 (crop management)#####
# scenario 2. the combination of measures with change in RES, CC, ROT (Optimized crop management vs

# make local copy
dt.s2 <- copy(dt.new)

# baseline mean and sd for total N input
dt.fert.bs <- dt.new[,list(mean = mean(nh4+no3+nam), sd = sd(nh4+no3+nam))]

# update actions taken for scenario 3
dt.s2[, fertilizer_typeenhanced := 0]
dt.s2[, fertilizer_typemineral := 1]
dt.s2[, fertilizer_typeorganic := 0]
dt.s2[, fertilizer_typecombined := 0]
dt.s2[, rfpyes := 0]
dt.s2[, rftyes := 0]
dt.s2[, rfryes := 0]
dt.s2[, crop_residueyes := 1]
dt.s2[, cover_crop_and_crop_rotationyes := 1]
dt.s2[, tillageno_till := 0]
dt.s2[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]
dt.s2[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 )*soc_scaled]
dt.s2[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.s2[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s2[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred.s2 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s2[,c(cols) := dt.pred.s2]

```

```

# compare baseline with scenario

# select relevant columns of the baseline
dt.fin <- dt.new[,.(x,y,base = pMDmean,cropname,area)]

# select relevant columns of scenario 1 and merge
dt.fin <- merge(dt.fin,dt.s2[,.(x,y,s2 = pMDmean,cropname)],by=c('x','y','cropname'))

# estimate relative improvement via senario 1
dt.fin[, improvement := s2 - base]

# estimate area weighted mean relative improvement
dt.fin <- dt.fin[,list(improvement = weighted.mean(improvement,w = area)),by = c('x','y')]

# make spatial raster of the estimated improvement

# convert to spatial raster
r.fin <- terra::rast(dt.fin,type='xyz')
terra::crs(r.fin) <- 'epsg:4326'

# write as output
terra::writeRaster(r.fin,'D:/coursework/tang/ /tif/scenario_2.tif', overwrite = TRUE)

##### scenario 3 (NT/RT) #####
# scenario 3. the combination of measures with change in NT/RT. (Optimized fertilizer strategy vs.C

# make local copy
dt.s3 <- copy(dt.new)

# baseline mean and sd for total N input
dt.fert.bs <- dt.new[,list(mean = mean(nh4+no3+nam), sd = sd(nh4+no3+nam))]

# update actions taken for scenario 1
dt.s3[, fertilizer_typeenhanced := 0]
dt.s3[, fertilizer_typemineral := 1]
dt.s3[, fertilizer_typeorganic := 0]
dt.s3[, fertilizer_typecombined := 0]
dt.s3[, rfpyes := 0]
dt.s3[, rftyes := 0]
dt.s3[, rfryes := 0]
dt.s3[, crop_residueyes := 0]
dt.s3[, cover_crop_and_crop_rotationyes := 0]
dt.s3[, `tillageno-till` := 1]
dt.s3[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]
dt.s3[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 )*soc_scaled]
dt.s3[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.s3[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s3[,mget(c(m1.cols))])

```

```

# predict the NUE via MD model
dt.pred.s3 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s3[,c(cols) := dt.pred.s3]

# compare baseline with scenario

# select relevant columns of the baseline
dt.fin <- dt.new[,.(x,y,base = pMDmean,cropname,area)]

# select relevant columns of scenario 1 and merge
dt.fin <- merge(dt.fin,dt.s3[,.(x,y,s3 = pMDmean,cropname)],by=c('x','y','cropname'))

# estimate relative improvement via senario 1
dt.fin[, improvement := s3 - base]

# estimate area weighted mean relative improvement
dt.fin <- dt.fin[,list(improvement = weighted.mean(improvement,w = area)),by = c('x','y')]

# make spatial raster of the estimated improvement

# convert to spatial raster
r.fin <- terra::rast(dt.fin,type='xyz')
terra::crs(r.fin) <- 'epsg:4326'

# write as output
terra::writeRaster(r.fin,'D:/coursework/tang/ /tif/scenario_3.tif', overwrite = TRUE)

```

```
#####
```

```

# plotting

library(ggplot2)
library(sf)

```

```
## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf_use_s2() is TRUE
```

```

library(rnaturalearth)
library(rnaturalearthdata)

```

```

##
## 'rnaturalearthdata'
## The following object is masked from 'package:rnaturalearth':
##
## countries110

```

```

library(terra)
library(cowplot)
library(vcd)

```

```

## grid
##
## 'grid'

```



```
## The following object is masked from 'package:terra':
```

```
##
```

```
##   depth
```

```
##
```

```
##   'vcd'
```

```
## The following objects are masked from 'package:terra':
```

```
##
```

```
##   mosaic, sieve
```

```
##### scenario_1 (optimal nutrient management) #####
```

```
# set theme
```

```
theme_set(theme_bw())
```

```
# get the raster to plot
```

```
r1 <- terra::rast('D:/coursework/tang/ /tif/scenario_1.tif')
```

```
# convert to data.frame
```

```
r1.p <- as.data.frame(r1,xy=TRUE)
```

```
# Exclude outliers greater than 70%
```

```
r1.p <- r1.p[r1.p$improvement < 70,]
```

```
#r1.p.mean <- paste0(round(mean(r1.p$improvement)), '%')
```

```
# get base world map
```

```
world <- ne_countries(scale = "medium", returnclass = "sf")
```

```
#plot a basic world map plot
```

```
p1 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +  
  geom_tile(data = r1.p, aes(x=x, y=y, name = 'none',  
                             fill = cut(improvement, breaks = c(15, 25, 30, 35, 800),  
                                         labels = c('< 25', '25-30', '30-35', '> 35')))) +
```

```
  # scale_fill_gradientn(colours = rainbow(3)) +
```

```
  #scale_fill_viridis_c()+
```

```
  theme_void() +
```

```
  theme(legend.position = c(0.05, 0.4), text = element_text(size = 12),
```

```
        legend.background = element_rect(fill = NA, color = NA),
```

```
        panel.border = element_blank()) +
```

```
  labs(fill = 'NUEr increased (%)') +
```

```
  xlab("Longitude") + ylab("Latitude") +
```

```
  ggtitle("Optimal nutrient management") +
```

```
  theme(plot.title = element_text(size = 16))+
```

```
  theme(plot.title = element_text(hjust = 0.5))+
```

```
  annotate("text", x=0.5, y=-50, label="Mean: 26.9%", size=5, colour="#0070C0", fontface = "bold")+
```

```
  coord_sf(crs = 4326)
```

```
## Warning in geom_tile(data = r1.p, aes(x = x, y = y, name = "none", fill =
```

```
## cut(improvement, : Ignoring unknown aesthetics: name
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
```

```
## 3.5.0.
```

```
## i Please use the `legend.position.inside` argument of `theme()` instead.
```

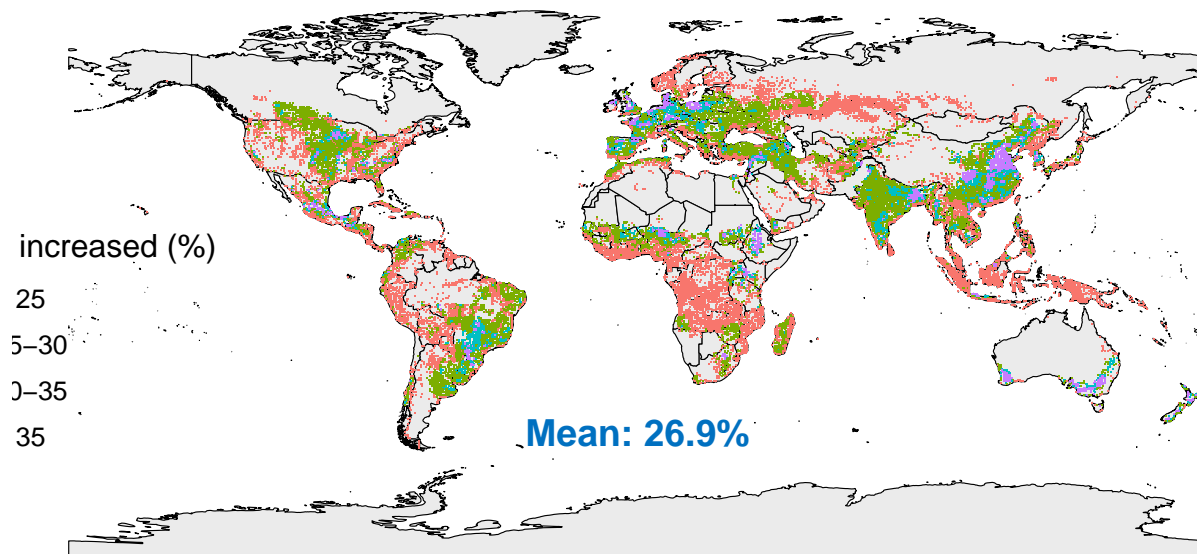
```
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.
```

```
p1
```

Optimal nutrient management



```
##### scenario_2 (optimal crop management) #####
```

```
# set theme
```

```
theme_set(theme_bw())
```

```
# get the raster to plot
```

```
r2 <- terra::rast('D:/coursework/tang/ /tif/scenario_2.tif')
```

```
# convert to data.frame
```

```
r2.p <- as.data.frame(r2,xy=TRUE)
```

```
# Exclude outliers greater than 70%
```

```
r2.p <- r2.p[r2.p$improvement < 70,]
```

```
#r2.p.mean <- paste0(round(mean(r2.p$improvement)), '%')
```

```
# get base world map
```

```
world <- ne_countries(scale = "medium", returnclass = "sf")
```

```
# plot a basic world map plot
```

```
p2 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +  
  geom_tile(data = r2.p, aes(x=x, y=y, name = 'none',  
    fill = cut(improvement, breaks = c(0, 5, 10, 15, 800),  
    labels = c('< 5', '5-10', '10-15', '> 15') ))) +
```

```

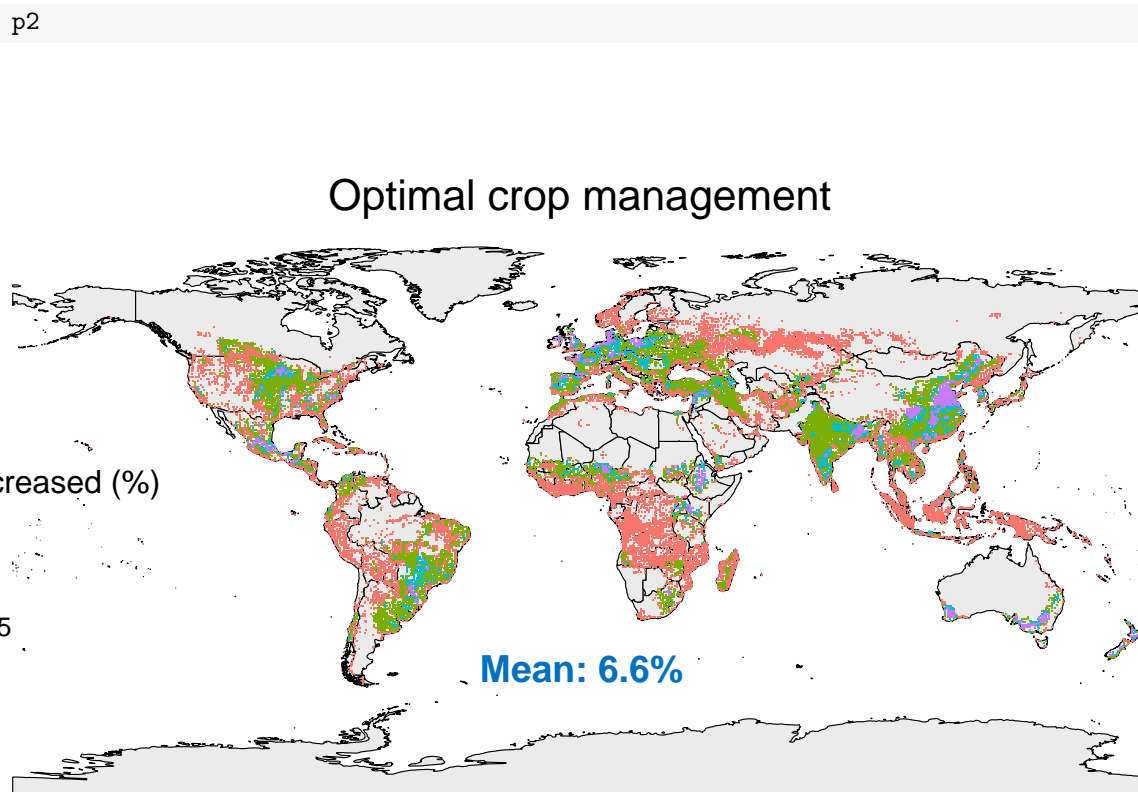
theme_void() +
theme(legend.position = c(0.05,0.4), text = element_text(size = 12),
      legend.background = element_rect(fill = NA,color = NA),
      panel.border = element_blank()) +
labs(fill = 'NUEr increased (%)') +
xlab("Longitude") + ylab("Latitude") +
ggtitle("Optimal crop management") +
theme(plot.title = element_text(size = 16))+
theme(plot.title = element_text(hjust = 0.5))+
annotate("text",x=0.5,y=-50,label="Mean: 6.6%",size=5, colour="#0070C0",fontface = "bold")+
coord_sf(crs = 4326)

```

```

## Warning in geom_tile(data = r2.p, aes(x = x, y = y, name = "none", fill =
## cut(improvement, : Ignoring unknown aesthetics: name

```



```

##### scenario_3 (optimal soil management) #####

# set theme
theme_set(theme_bw())

# get the raster to plot
r3 <- terra::rast('D:/coursework/tang/ /tif/scenario_3.tif')

# convert to data.frame
r3.p <- as.data.frame(r3,xy=TRUE)

```

```

# Exclude outliers greater than 70%
r3.p <- r3.p[r3.p$improvement <70,]
#r3.p.mean <- paste0(round(mean(r3.p$improvement)), '%')

# write.csv(r3.p, file="E:/phD/Papers/paper2/You_et_al_2022/NC/tillage.csv")

# get base world map
world <- ne_countries(scale = "medium", returnclass = "sf")

# plot a basic world map plot
p3 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +
  geom_tile(data = r3.p, aes(x=x, y=y, name = 'none',
                             fill = cut(improvement, breaks= c(-7,0,5,10,800),
                                         labels = c('< 0', '0-5', '5-10', '> 10') ))) +

  theme_void() +
  theme(legend.position = c(0.05,0.4), text = element_text(size = 12),
        legend.background = element_rect(fill = NA, color = NA),
        panel.border = element_blank()) +
  labs(fill = 'NUEr increased (%)') +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Optimal soil management") +
  theme(plot.title = element_text(size = 16))+
  theme(plot.title = element_text(hjust = 0.5))+
  annotate("text", x=0.5, y=-50, label="Mean: 0.6%", size=5, colour="#0070C0", fontface = "bold")+
  coord_sf(crs = 4326)

```

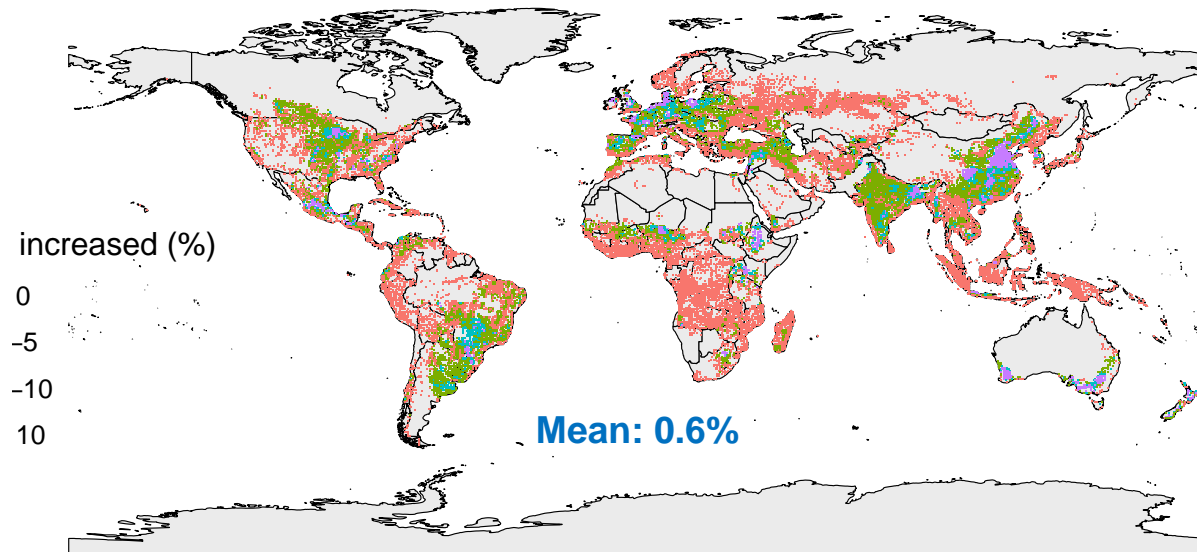
```

## Warning in geom_tile(data = r3.p, aes(x = x, y = y, name = "none", fill =
## cut(improvement, : Ignoring unknown aesthetics: name

```

```
p3
```

Optimal soil management



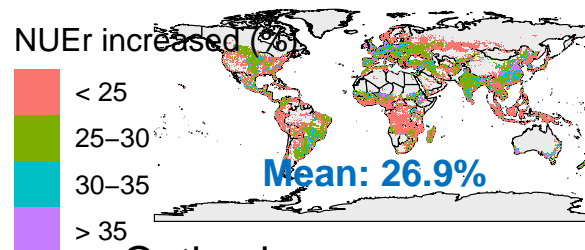
```
#2*2
```

```
library(ggpubr)
```

```
##  
## 'ggpubr'  
## The following object is masked from 'package:cowplot':  
##  
## get_legend  
## The following object is masked from 'package:terra':  
##  
## rotate  
  
p<-ggarrange(p1, p2, p3, ncol = 1, nrow = 3, #common.legend = TRUE, legend = "bottom",  
             labels = c("a", "b", "c"), font.label=list(size=14),hjust = 0, vjust = 1)  
  
p
```

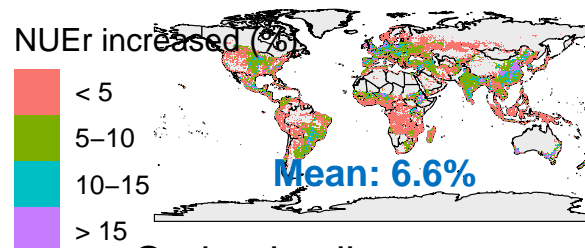
a

Optimal nutrient management



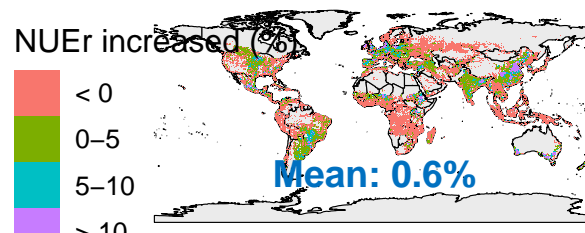
b

Optimal crop management



c

Optimal soil management



```
ggsave(p, file = "D:/coursework/tang/ /picture/Figure_5.png",width = 183,height = 247, units = "mm")
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

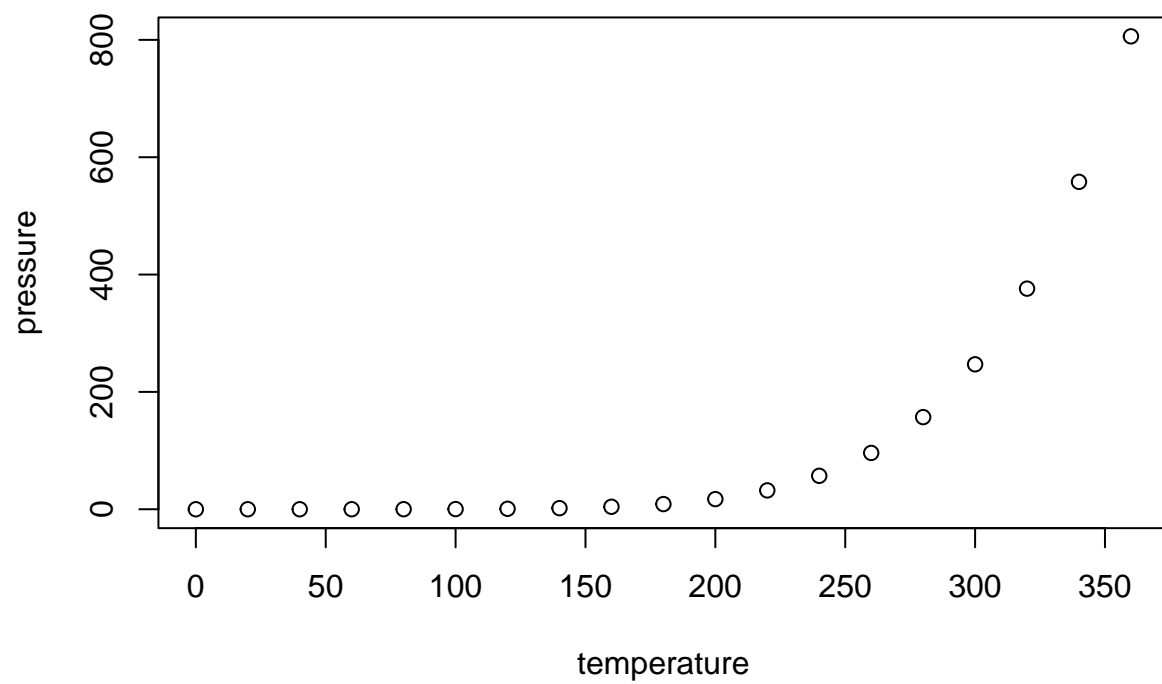
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.