

figure.4

wang xiaoqian

2025-05-20

```
# Figure 4
```

```
# require packages
```

```
require(terra)
```

```
## 载入需要的程序包: terra
```

```
## terra 1.8.42
```

```
require(data.table)
```

```
## 载入需要的程序包: data.table
```

```
##
```

```
## 载入程序包: 'data.table'
```

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

```
##
```

```
##      shift
```

```
require(metafor)
```

```
## 载入需要的程序包: metafor
```

```
## 载入需要的程序包: Matrix
```

```
## 载入需要的程序包: metadat
```

```
## 载入需要的程序包: numDeriv
```

```
##
```

```
## Loading the 'metafor' package (version 4.8-0). For an
```

```
## introduction to the package please type: help(metafor)
```

```
setwd("D:/date/homework")
```

```
# empty environment
```

```

rm(list=ls())

# load in cropland area Copernicus (not used)
if(FALSE){

  # source: https://www.esa-landcover-cci.org/?q=node/164)
  cropland <- terra::rast('data/cropland83km_v2.tif')

  cropland.dt <- as.data.table(cropland)
  setnames(cropland.dt, 'cci_croptype')

  # describe crops
  cropland.dt[cci_croptype == 10, ctype := 'rainfed_cropland']
  cropland.dt[cci_croptype == 20, ctype := 'irrigated_cropland']
  cropland.dt[cci_croptype == 30, ctype := 'mosaic_cropland']
  cropland.dt[cci_croptype == 40, ctype := 'natural_vegetation']
}

require(terra)
require(data.table)
require(metafor)

# Luncheng add

# read the raster with climatic data
r.clim <- terra::rast('data/climate.tif')

# read the raster with cropping data
r.crop <- rast('data/ma_crops.tif')

# read the raster with soil data
r.soil <- terra::rast('data/soil.tif')

# read the earlier prepared file with tillage practices
r.till <- terra::rast('data/tillage.tif')

# read the earlier prepared file with manure N dose
r.nman <- terra::rast('data/nofert.tif')

```

```

# read the earlier prepared file with fertilizer N dose
r.nfert <- terra::rast('data/nifert.tif')

# read the earlier prepared file with cropping intensity
r.cropint <- terra::rast('data/cropintensity.tif')

#####
# add all rasters

# clear environment
rm(list= ls())

# what rasters are in data
rfiles <- list.files('data', pattern = 'tif$',full.names = TRUE)
rfiles <- rfiles[!grepl('cropland|climzone|soc',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_
                        'nifert_nfert_nh4','nifert_nfert_no3','nofert_nofert','cropintensity_cro
                        new = c('mat','pre','phw','clay','soc','nh4','no3','nam','cropintensity'),skip_absent

# 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_ma_crops_RICE) & is.na(tillage_ma_crops_MAIZ) & is.na(tillage_ma_crops_other) & is.na(tillage_ma_crops_maize)) & !is.na(cropintensity)]
r.dt <- r.dt[!(is.na(tillage_RICE) & is.na(tillage_MAIZ) & is.na(tillage_other) & is.na(tillage_ma_crops_RICE) & is.na(tillage_ma_crops_MAIZ) & is.na(tillage_ma_crops_other) & is.na(tillage_ma_crops_maize)) & !is.na(cropintensity)]

# 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','croptype'),
                  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[,cropname := 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:/date/homework/Source Data.xlsx',sheet = "Figure4")
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 +
                   n_dose_scaled:soc_scaled + ctm:rfp + ctm + ctw + ctr + ctm:mat_scaled +
                   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
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
dt.new[, fertilizer_typeenhanced := 0]
dt.new[, fertilizer_typemineral := 1]
dt.new[, fertilizer_typeorganic := 0]
dt.new[, fertilizer_typecombined := 0]
dt.new[, rfpyes := 0]
dt.new[, rftyes := 0]
dt.new[, rfryes := 0]
dt.new[, crop_residueyes := 0]
dt.new[, cover_crop_and_crop_rotationyes := 0]
dt.new[, cover_crop_and_crop_rotationyes := fifelse(cropintensity>1,1,0)]
dt.new[, `tillageno-till` := fifelse(till_name == 'no-till',1,0)]
#dt.new[, `tillageno-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 4 (combination_mean)#####
# scenario 4. the combination of measures with change in EE, CF, RFR, RFT, BC, RES, CC, ROT

# make local copy
dt.s4 <- 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.s4[, fertilizer_typeenhanced := 1]
dt.s4[, fertilizer_typemineral := 0]
dt.s4[, fertilizer_typeorganic := 1]
dt.s4[, fertilizer_typecombined := 1]
dt.s4[, rfpyes := 1]
dt.s4[, rftyes := 1]
dt.s4[, rfryes := 1]
dt.s4[, crop_residueyes := 1]
dt.s4[, cover_crop_and_crop_rotationyes := 1]
dt.s4[, tillageno_till := 1]
dt.s4[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]
dt.s4[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1)*soc_scaled]
dt.s4[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.s4[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s4[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred.s4 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s4[,c(cols) := dt.pred.s4]

# 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.s4[,.(x,y,s4 = pMDmean,cropname)],by=c('x','y','cropname'))

# estimate relative improvement via senario 1
dt.fin[, improvement := s4 - 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:/date/homework/tif/scenario_4.tif', overwrite = TRUE)

##### scenario 5 (combination_lower boundaries of 95%
# make local copy
dt.s5 <- 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.s5[, fertilizer_typeenhanced := 1]
dt.s5[, fertilizer_typemineral := 0]
dt.s5[, fertilizer_typeorganic := 1]
dt.s5[, fertilizer_typecombined := 1]
dt.s5[, rfpyes := 1]
dt.s5[, rftyes := 1]
dt.s5[, rfryes := 1]
dt.s5[, crop_residueyes := 1]
dt.s5[, cover_crop_and_crop_rotationyes := 1]
dt.s5[, tillageno_till := 1]
dt.s5[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]

```

```

dt.s5[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 )*soc_scaled]
dt.s5[, `rfpyes:ctmyes` := rfpyes*ctmyes]
dt.s5[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s5[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred.s5 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s5[,c(cols) := dt.pred.s5]

# compare baseline with scenario

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

# select relevant columns of scenario 1 and merge
dt.fin <- merge(dt.fin,dt.s5[,.(x,y,s5_mean = pMDmean,s5_se = pMDse,cropname)],by=c('x','y','c

# calcualte the lower boundaries
dt.fin[, se_improvement := sqrt(s5_se^2 + base_se^2)]
dt.fin[, mean_improvement := s5_mean - base_mean]
dt.fin[, lower_improvement := mean_improvement - qnorm(0.975) * se_improvement]

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

# 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:/date/homework/tif/scenario_5.tif', overwrite = TRUE)

##### scenario 6 (combination_upper boundaries of 95%
# make local copy
dt.s6 <- 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.s6[, fertilizer_typeenhanced := 1]
dt.s6[, fertilizer_typemineral := 0]
dt.s6[, fertilizer_typeorganic := 1]
dt.s6[, fertilizer_typecombined := 1]
dt.s6[, rfpyes := 1]
dt.s6[, rftyes := 1]
dt.s6[, rfryes := 1]
dt.s6[, crop_residueyes := 1]
dt.s6[, cover_crop_and_crop_rotationyes := 1]
dt.s6[, tillageno_till := 1]
dt.s6[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean) / dt.fert.bs$sd ]
dt.s6[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 ) * soc_scaled]
dt.s6[, `rfpyes:ctmyes` := rfpyes * ctmyes]
dt.s6[, `mat_scaled:ctmyes` := mat_scaled * ctmyes]

# convert to matrix, needed for rma models
dt.newmod <- as.matrix(dt.s6[,mget(c(m1.cols))])

# predict the NUE via MD model
dt.pred.s6 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
dt.s6[,c(cols) := dt.pred.s6]

# compare baseline with scenario

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

# select relevant columns of scenario 1 and merge
dt.fin <- merge(dt.fin,dt.s6[,.(x,y,s6_mean = pMDmean,s6_se = pMDse,cropname)],by=c('x','y','cropname'))

# calculate the lower boundaries
dt.fin[, se_improvement := sqrt(s6_se^2 + base_se^2)]
dt.fin[, mean_improvement := s6_mean - base_mean]
dt.fin[, upper_improvement := mean_improvement + qnorm(0.975) * se_improvement]

```

```

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

# 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:/date/homework/tif/scenario_6.tif', overwrite = TRUE)

# ##### scenario 4 (combination_SE)#####
# # scenario 4. the combination of measures with change in EE, CF, RFR, RFT, BC, RES, CC, ROT
#
# # make local copy
# dt.s5 <- 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.s5[, fertilizer_typeenhanced := 1]
# dt.s5[, fertilizer_typemineral := 0]
# dt.s5[, fertilizer_typeorganic := 1]
# dt.s5[, fertilizer_typecombined := 1]
# dt.s5[, rfpyes := 1]
# dt.s5[, rftyes := 1]
# dt.s5[, rfryes := 1]
# dt.s5[, crop_residueyes := 1]
# dt.s5[, cover_crop_and_crop_rotationyes := 1]
# dt.s5[, tillageno_till := 1]
# dt.s5[, n_dose_scaled := ((nh4+no3+nam) * 0.7 - dt.fert.bs$mean)/ dt.fert.bs$sd ]
# dt.s5[, `n_dose_scaled:soc_scaled` := (n_dose_scaled - 0.1 )*soc_scaled]
# dt.s5[, `rfpyes:ctmyes` := rfpyes*ctmyes]
# dt.s5[, `mat_scaled:ctmyes` := mat_scaled*ctmyes]
#
# # convert to matrix, needed for rma models

```

```

# dt.newmod <- as.matrix(dt.s5[,mget(c(m1.cols))])
#
# # predict the NUE via MD model
# dt.pred.s5 <- as.data.table(predict(m1,newmods = dt.newmod,addx=F))
# dt.s5[,c(cols) := dt.pred.s5]
#
# # compare baseline with scenario
#
# # select relevant columns of the baseline
# dt.fin <- dt.new[,.(x,y,base_mean = pMDmean, base_se = pMDse,cropname,area)]
#
# # select relevant columns of scenario 1 and merge
# dt.fin <- merge(dt.fin,dt.s5[,.(x,y,s5_mean = pMDmean,s5_se = pMDse,cropname)],by=c('x','y',
#
# # estimate relative improvement via senario 1
# dt.fin[, se_improvement := sqrt(s5_se^2 + base_se^2)]
#
# # # calcualte the CV, assuming n = 1
# # dt.fin[, mean_improvement := s5_mean - base_mean]
# # dt.fin[, cv_improvement := round(se_improvement * sqrt(1) * 100 / mean_improvement,0)]
#
# # estimate area weighted mean relative improvement
# dt.fin <- dt.fin[,list(se_improvement = weighted.mean(se_improvement,w = area)),by = c('x','y')]
# #dt.fin <- dt.fin[,list(cv_improvement = weighted.mean(cv_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,'C:/Users/86188/Desktop/Figures/scenario_5.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)
```

```
##
## 载入程序包: 'rnaturalearthdata'
## The following object is masked from 'package:rnaturalearth':
##
##      countries110
```

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

```
## 载入需要的程序包: grid
##
## 载入程序包: 'grid'
## The following object is masked from 'package:terra':
##
##      depth
##
## 载入程序包: 'vcd'
## The following objects are masked from 'package:terra':
##
##      mosaic, sieve
```

```
library(ggpubr)
```

```
##
## 载入程序包: 'ggpubr'
## The following object is masked from 'package:cowplot':
##
##      get_legend
## The following object is masked from 'package:terra':
##
##      rotate
```

```
##### scenario_4 (combined_mean) #####

# set theme
theme_set(theme_bw())

# get the raster to plot
r4 <- terra::rast('D:/date/homework/tif/scenario_4.tif')

# convert to data.frame
r4.p <- as.data.frame(r4,xy=TRUE)

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

# plot a basic world map plot
p4 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +
  geom_tile(data = r4.p, aes(x=x, y=y, name = 'none',
                             fill = cut(improvement, breaks= c(0,20,30,35,40,800),
                             labels = c('< 20', '20-30', '30-35', '35-40', '> 40') ))) +
  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()) +
  scale_fill_manual(values = c("#FF7F50", "#a3a500", "#00bf7d", "#00b0f6", "#e76bf3"), drop=FALSE)
  labs(fill = 'NUEr increased (%)') +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Mean absolute NUEr changes") +
  theme(plot.title = element_text(size = 16)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  annotate("text", x=0.5, y=-50, label="Mean: 30.3%", size=5, colour="#0070C0", fontface = "bold") +
  coord_sf(crs = 4326)
```

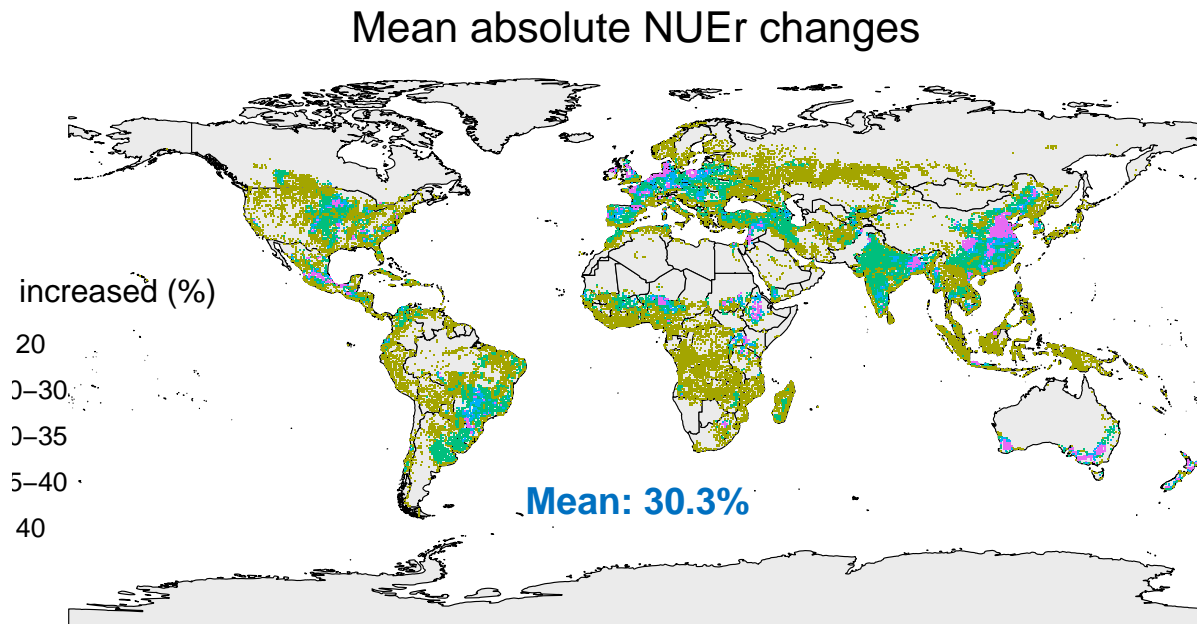
```
## Warning in geom_tile(data = r4.p, aes(x = x, y = y, name = "none", fill =
## cut(improvement, : Ignoring unknown aesthetics: name

## 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.
```

```
p4
```



```
##### scenario_5 (combined_lower boundary) #####
```

```
# set theme
```

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

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

```
r5 <- terra::rast('D:/date/homework/tif/scenario_5.tif')
```

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

```
r5.p <- as.data.frame(r5,xy=TRUE)
```

```
# get base world map
```

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

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

```
p5 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +
```



```

geom_tile(data = r5.p,aes(x=x,y=y, name = 'none',
                        fill = cut(lower_improvement,breaks= c(0,20,30,35,40,800),
                        labels = c('< 20','20-30','30-35','35-40','> 40') ))) +

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()) +
scale_fill_manual(values = c("#FF7F50", "#a3a500", "#00bf7d", "#00b0f6", "#e76bf3"),drop=FALSE)
labs(fill = 'NUEr increased (%)') +
xlab("Longitude") + ylab("Latitude") +
ggtitle("Lower boundaries of 95% CI for NUEr changes") +
theme(plot.title = element_text(size = 16))+
theme(plot.title = element_text(hjust = 0.5))+
annotate("text",x=0.5,y=-50,label="Mean: 24.5%",size=5, colour="#0070C0",fontface = "bold")+
coord_sf(crs = 4326)

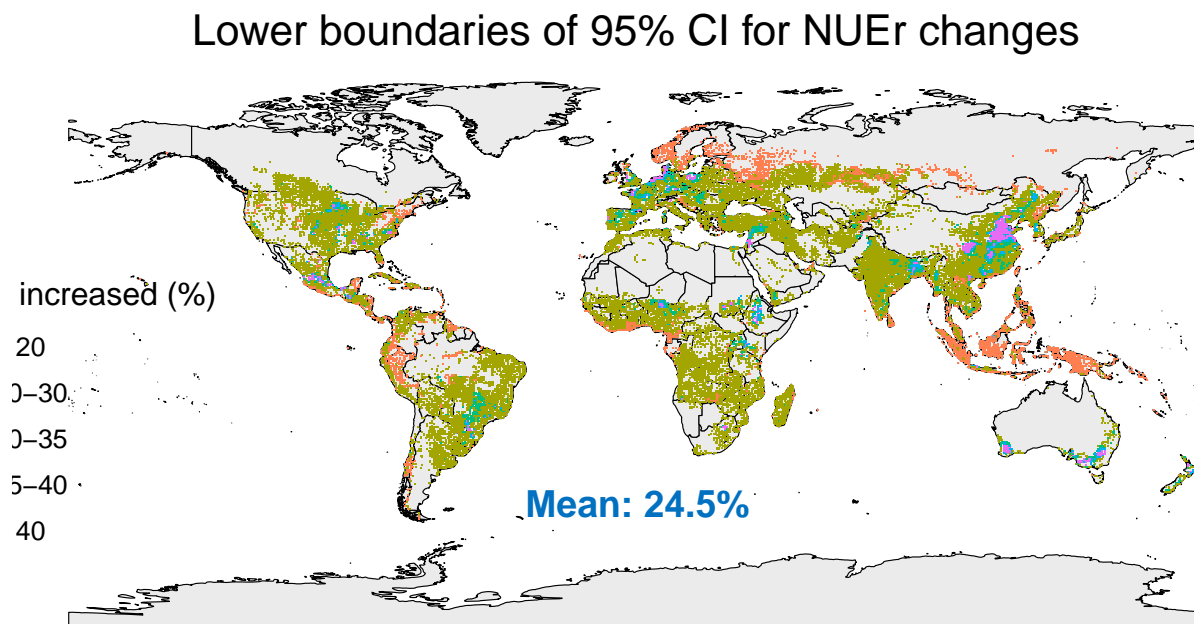
```

```

## Warning in geom_tile(data = r5.p, aes(x = x, y = y, name = "none", fill =
## cut(lower_improvement, : Ignoring unknown aesthetics: name

```

p5



```
##### scenario_6 (combined_lower boundary) #####

# set theme
theme_set(theme_bw())

# get the raster to plot
r6 <- terra::rast('D:/date/homework/tif/scenario_6.tif')

# convert to data.frame
r6.p <- as.data.frame(r6,xy=TRUE)

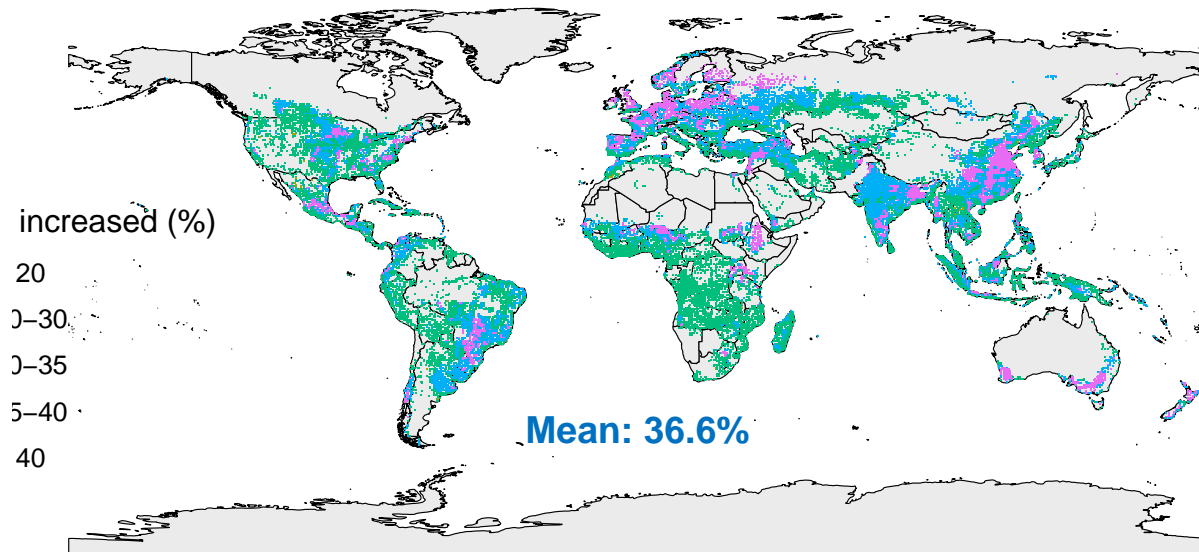
# get base world map
world <- ne_countries(scale = "medium", returnclass = "sf")

# plot a basic world map plot
p6 <- ggplot(data = world) + geom_sf(color = "black", fill = "gray92") +
  geom_tile(data = r6.p, aes(x=x, y=y, name = 'none',
                             fill = cut(upper_improvement, breaks= c(0,20,30,35,40,800),
                             labels = c('< 20', '20-30', '30-35', '35-40', '> 40') ))) +
  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()) +
  scale_fill_manual(values = c("#FF7F50", "#a3a500", "#00bf7d", "#00b0f6", "#e76bf3"), drop=FALSE)
  labs(fill = 'NUEr increased (%)') +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Upper boundaries of 95% CI for NUEr changes") +
  theme(plot.title = element_text(size = 16)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  annotate("text", x=0.5, y=-50, label="Mean: 36.6%", size=5, colour="#0070C0", fontface = "bold") +
  coord_sf(crs = 4326)
```

```
## Warning in geom_tile(data = r6.p, aes(x = x, y = y, name = "none", fill =
## cut(upper_improvement, : Ignoring unknown aesthetics: name
```

```
p6
```

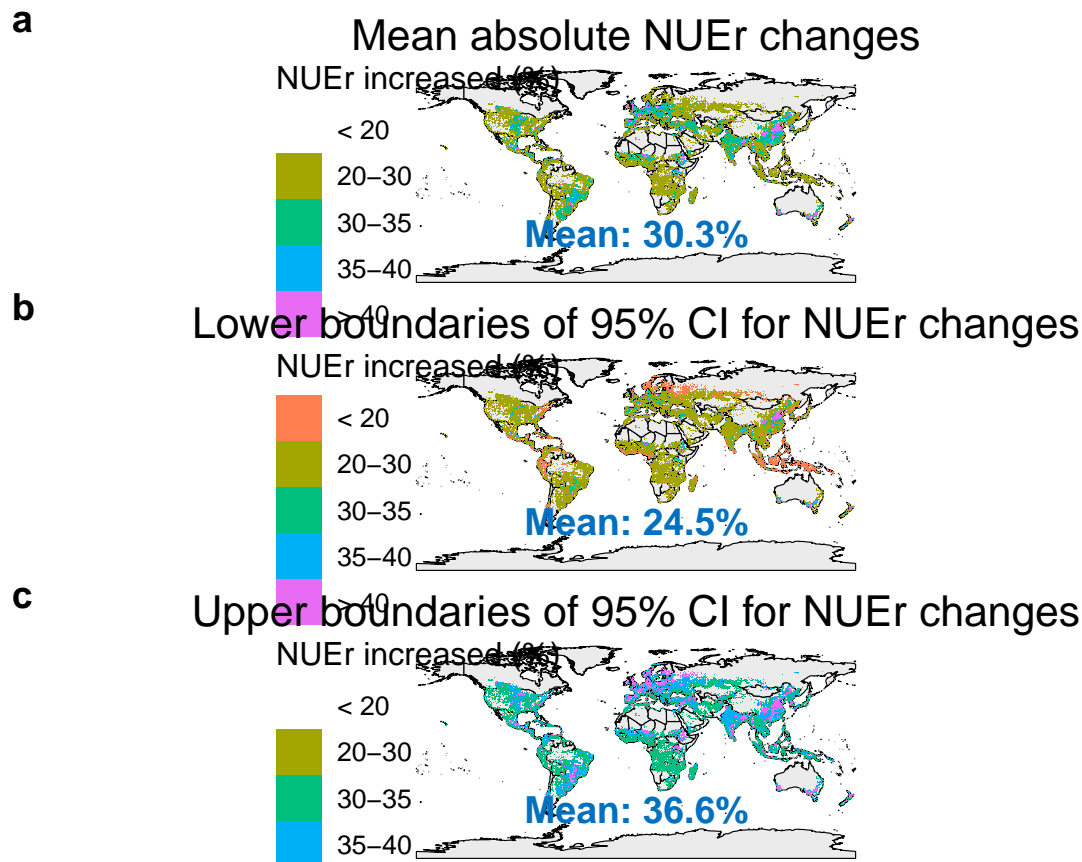
Upper boundaries of 95% CI for NUEr changes



```
library(ggpubr)
```

```
p<-ggarrange(p4, p5, p6, ncol = 1, nrow = 3, labels = c("a", "b", "c"), font.label=list(size=14
```

```
p
```



```
ggsave(p, file = "D:/date/homework/picture/Figure_4.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>.

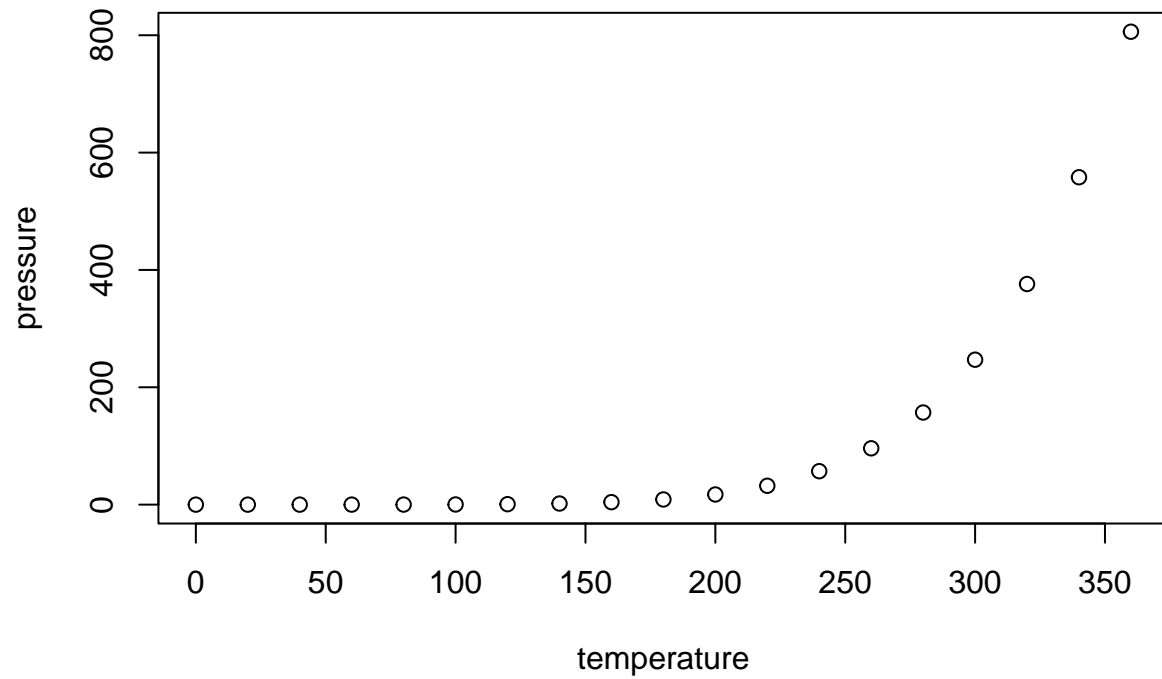
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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.