ghana presentation tables

Tomas Morley

20 april 2017

## table 1

###### ANALYSIS of GHA panel data

# CHECK:

# GEO AND AREA TO RUN FROM ETHYG FOLDER!!

# Imputation of dummy variables.

# Mean scale variables (see Henningsen)

# SFA from other package

# Add texture to soil variables

PACKAGES ETC

library(pacman) p\_load(char=c("dplyr", "rprojroot", "stargazer", "frontier", "moments", "AER"), install=TRUE) Error in (function (n) : lazy-load database 'C:/Program Files/R/R-3.3.2/library/Matrix/R/Matrix.rdb' is corrupt Failed with error: ‘package ‘car’ could not be loaded’ trying URL '<https://cran.rstudio.com/bin/windows/contrib/3.3/AER_1.2-5.zip>' Content type 'application/zip' length 2457971 bytes (2.3 MB) downloaded 2.3 MB

package ‘AER’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in C:00126i\_packages

AER installed Error in (function (n) : lazy-load database 'C:/Program Files/R/R-3.3.2/library/Matrix/R/Matrix.rdb' is corrupt Failed with error: ‘package ‘car’ could not be loaded’ > root <- find\_root(is\_rstudio\_project) > setwd(root) > > source("Code/winsor.r") > source("Code/predict.sfa.R") > options(scipen=999) > > ####################################### > ############## LOAD DATA ############## > ####################################### > > # Load 2010 data > dbP <- readRDS("Cache/GHA2010.rds") > > ####################################### > ############## CLEANING ############### > ####################################### > > > # Create id for plots > dbP <- dbP %>% + mutate(id=1:dim(.)[1]) > > # Cleaning and analysis depends strongly on which measure is chosen for area, which is the denominator for many variables. > # there are three possible yield variables. > # that can be created for the last two waves of data. > # 1. yld1: above uses the full gps areas as denominator > # 2. yld2: uses harvested area as denominator > # 3. yld3: Uses relative harvest area to correct gps area > # To simplify the code we set these values in this part. Subsequent analysis code can then be used for any definition of yield. > > # GHA does not present information on area harvested, only plot size. Area is only farmer self-assessed. GPS is not presented. > # We use area here but definition is comparable to yld1 > > dbP <- dbP %>% + mutate( + area = area, + #area = area\_gps, + yld = crop\_qty\_harv/area, + N = N/area) > > > # As we focus on small scale farmers we restrict area size > dbP <- filter(dbP, area <=10) > > # cap yield at 18593 kg/ha, the highest potential yield in ETH (not water limited) > dbP <- filter(dbP, yld <= 15229.37956) > > # Restrict attention to plots that use N < 700. 700kg/hectare represents an upper bound limit associated with inorganic fertilizer use in the United States under irrigated corn conditions (Sheahan & Barett 2014) > dbP <- filter(dbP, N < 700) > > # Filter out plots with zero labour > # NOTE we only have harvest labour > # CHECK might replace this with adult equivalent variable! > dbP <- dbP %>% + mutate(lab = lab\_val1 + lab\_val2 + lab\_val3 + lab\_val4) %>% + filter(lab >0) %>% + select(-lab) > > > # CHECK ADD SOIL VARIABLES > # Select relevant variables and complete cases > db0 <- dbP %>% + dplyr::select(hhid, id, ZONE, REGNAME, plotno, # ZONE AND REGNAMES reversed to remain consistent with other LSMS + AEZ, fs, + SOC, SOC2, ph, ph2, RootDepth, + #soildepth, soiltype, + #rain\_year, rain\_wq, + #SPEI, + #YA, YW, YP, + elevation, + #nutr\_av, + yld, + crop\_qty\_harv, + #sold\_qty\_kg, sold\_qty\_gr, + lab\_val1, lab\_val2, lab\_val3, lab\_val4, + implmt\_valu, lvstk\_valu, + fung, herb, insec, manure, mech, + #seed\_type, + N, + irrig, + area, area\_tot, + #sex, age, + #ed\_any, family\_size, credit, + #literate, cage, death, N1555, + dist\_hh, + #trans\_cost, + #title, + #popEA, + #extension, extension2, + #fert\_source, + #road, cost2small\_town, bank, micro\_finance, ext\_agent, + crop\_count, + rural, + lat, lon, YW) > > > summary(db0) hhid id ZONE REGNAME plotno  
Length:1500 Min. : 1 Min. :1.00 Length:1500 Min. :1.00  
Class :character 1st Qu.: 385 1st Qu.:1.00 Class :character 1st Qu.:1.00  
Mode :character Median : 772 Median :2.00 Mode :character Median :1.00  
Mean : 778 Mean :1.64 Mean :1.46  
3rd Qu.:1170 3rd Qu.:2.00 3rd Qu.:2.00  
Max. :1553 Max. :3.00 Max. :5.00

AEZ fs SOC SOC2

Tropic - warm / semiarid: 9 Artisanal fishing :120 Min. : 1.14 Min. : 1.53  
Tropic - warm / subhumid:1021 Cereal-root crop mixed :498 1st Qu.: 2.67 1st Qu.: 4.75  
Tropic - warm / humid : 377 Humid lowland tree crop:411 Median : 4.01 Median : 6.93  
NA's : 93 Irrigated : 4 Mean : 4.19 Mean : 7.55  
Root and tuber crop :378 3rd Qu.: 5.33 3rd Qu.: 9.03  
NA's : 89 Max. :13.33 Max. :30.77  
NA's :92 NA's :92  
ph ph2 RootDepth elevation yld crop\_qty\_harv  
Min. :49.0 Min. :48.9 Min. : 0.0 Min. : -5.4 Min. : 1 Min. : 4  
1st Qu.:56.7 1st Qu.:56.1 1st Qu.: 52.8 1st Qu.:133.7 1st Qu.: 207 1st Qu.: 209  
Median :60.2 Median :60.1 Median : 73.6 Median :190.7 Median : 483 Median : 419  
Mean :59.4 Mean :59.3 Mean : 73.9 Mean :196.5 Mean : 726 Mean : 819  
3rd Qu.:61.9 3rd Qu.:61.8 3rd Qu.: 98.0 3rd Qu.:251.7 3rd Qu.: 920 3rd Qu.: 838  
Max. :72.9 Max. :75.1 Max. :133.1 Max. :507.2 Max. :13318 Max. :18147  
NA's :92 NA's :92 NA's :93 NA's :89  
lab\_val1 lab\_val2 lab\_val3 lab\_val4 implmt\_valu lvstk\_valu  
Min. : 1.0 Min. : 22 Min. : 8.0 Min. : 20 Min. : 0 Min. : 0  
1st Qu.: 36.0 1st Qu.:211 1st Qu.: 47.0 1st Qu.:173 1st Qu.: 12 1st Qu.: 0  
Median : 49.0 Median :322 Median : 63.0 Median :274 Median : 23 Median : 45  
Mean : 53.1 Mean :300 Mean : 69.3 Mean :248 Mean : 114 Mean : 584  
3rd Qu.: 64.0 3rd Qu.:367 3rd Qu.: 84.0 3rd Qu.:323 3rd Qu.: 46 3rd Qu.: 350  
Max. :400.0 Max. :879 Max. :357.0 Max. :569 Max. :21002 Max. :37805

fung herb insec manure mech

Min. :0.000 Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.000  
1st Qu.:0.000 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000  
Median :0.000 Median :0.000 Median :0.0000 Median :0.0000 Median :0.000  
Mean :0.004 Mean :0.198 Mean :0.0113 Mean :0.0553 Mean :0.289  
3rd Qu.:0.000 3rd Qu.:0.000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.000  
Max. :1.000 Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.000

N irrig area area\_tot dist\_hh

Min. : 0.00 Min. :0.0000 Min. :0.020 Min. : 0.08 Length:1500  
1st Qu.: 0.00 1st Qu.:0.0000 1st Qu.:0.607 1st Qu.: 1.21 Class :character  
Median : 0.00 Median :0.0000 Median :1.214 Median : 2.23 Mode :character  
Mean : 4.88 Mean :0.0387 Mean :1.415 Mean : 2.96  
3rd Qu.: 0.00 3rd Qu.:0.0000 3rd Qu.:1.619 3rd Qu.: 3.71  
Max. :185.32 Max. :1.0000 Max. :9.010 Max. :62.73  
NA's :2  
crop\_count rural lat lon YW  
Min. :1.00 Urban: 132 Min. : 4.88 Min. :-2.81 Min. : 4.55  
1st Qu.:1.00 Rural:1368 1st Qu.: 6.39 1st Qu.:-1.67 1st Qu.: 7.92  
Median :2.00 Median : 7.38 Median :-0.62 Median : 8.09  
Mean :2.39 Mean : 7.88 Mean :-0.79 Mean : 8.41  
3rd Qu.:3.00 3rd Qu.: 9.56 3rd Qu.:-0.01 3rd Qu.: 8.41  
Max. :5.00 Max. :11.13 Max. : 1.15 Max. :11.11  
NA's :89 NA's :89 NA's :247  
> > > ####################################### > ###### COMPLETE CASES DATABASE ######## > ####################################### > > db0 <- db0 %>% + do(filter(., complete.cases(.))) > > > ###################################### > ######## Modify and add variables #### > ###################################### > > > # Following Burke > db0ph < 55] <- 1 > db0ph >= 55 & db0$ph <=70] <- 2 # Neutral and best suited for crops > db0$phdum[db0$ph > 70] <- 3 > db0phdum) > > > #db0ph2 < 55] <- 1 > db0ph2 >= 55 & db0$ph2 <=70, 1, 0) > #db0$phdum2[db0$ph2 >= 55 & db0$ph2 <=70] <- 1 > #db0ph2 > 70] <- 3 > #db0phdum2) > > # Recode AEZ into 4 zones > # db0AEZ > # db0AEZ2, from = c("Tropic-warm / semi-arid"), to = c("Tropic-warm")) > # db0AEZ2, from = c("Tropic-warm / sub-humid"), to = c("Tropic-warm")) > # db0AEZ2) > > # Crop count > 1 > db0crop\_count==1] <- 1 > db0crop\_count>1] <- 0 > > # additional variables > db0 <- db0 %>% mutate (logyld=log(yld), + yesN = ifelse(N>0, 1,0), # Dummy when plot does not use fertilizer, following approach of Battese (1997) + noN = ifelse(N<=0, 1,0), # Dummy when plot does use fertilizer, following approach of Battese (1997) + logN = log(pmax(N, noN)), # maximum of dummy and N following Battese (1997) + lab = lab\_val1 + lab\_val2 + lab\_val3 + lab\_val4, + #hirelab\_sh = harv\_lab\_hire/(harv\_lab\_hire + harv\_lab)*100, + #dumoxen = ifelse(oxen>0, 1,oxen), + lab=lab/area, + #logae = log(ae), + #asset = implmt\_value + lvstk2\_valu, + #assetph=asset/area\_tot, + #logasset = log(assetph+1), + loglab = log(lab), + logarea = log(area), # area\_gps not area because we want to add plot size as proxy for economies of scale + asset = (implmt\_valu +lvstk\_valu)/area\_tot, + logasset = log(asset+1) + #rain\_wq2 = rain\_wq*rain\_wq, + #pestherb = ifelse(herb==1 | pest==1, 1, 0), + #ext = ifelse(ext\_dummy\_pp==1 | ext\_dummy\_ph ==1, 1, 0), + #lograin = log(rain\_year), + #dumfertsource = recode(fert\_source, c("'Government' = 1; else = 0")), + #surveyyear2 = replace(surveyyear==2011, 1, 0) + ) > > # Add Translog variables > db0 <- db0 %>% + mutate(logNsq = logN^2, + loglabsq = loglab^2, + logassetsq = logasset^2, + logNlab = logN \* loglab, + logNasset = logN \* logasset, + loglabasset = loglab \* logasset + ) > > db0 <- droplevels(db0) > > > ###################################### > ##### Get plot specific pricess ###### > ###################################### > > db1 <- db0 > # Load and merge price data > Prices <- readRDS("cache/Prices\_GHA.rds") > > # Merge with panel data > db1 <- left\_join(db0, Prices) %>% + mutate(relprice = Pns/Pc) %>% + rename(Pm = Pc) Joining, by = c("hhid", "ZONE", "REGNAME", "plotno") > > # Drop unused levels (e.g. Zanzibar in zone), which are giving problems with sfa > db1 <- droplevels(db1) > db1N > 0, 1, 0) > > # summary stats table > # make a summary tab of variables for paper > sum\_dat <- select(db1, yld, N, area, lab, asset, + herb, mech, elevation, SOC2, + phdum2, yesN) > > > Mean <- colMeans(sum\_dat, na.rm=TRUE) > Median <- apply(sum\_dat, 2, function(col) median(col, na.rm=TRUE)) > SD <- apply(sum\_dat, 2, function(col) sd(col, na.rm=TRUE)) > Skewness <- apply(sum\_dat, 2, function(col) skewness(col, na.rm=TRUE)) > Min <- apply(sum\_dat, 2, function(col) min(col, na.rm=TRUE)) > Max <- apply(sum\_dat, 2, function(col) max(col, na.rm=TRUE)) > sum\_tab <- data.frame(Variable=colnames(sum\_dat), + Mean, Median, SD, Skewness, + Min, Max) > sum\_tab[, -1] <- round(sum\_tab[,-1], 3) > row.names(sum\_tab) <- NULL > > saveRDS(sum\_tab, "C:/users/morle001/WEcR/GHAYG/Cache/sum\_tab\_res.rds") > stargazer(select(db1, yld, area, crop\_count) %>% data.frame, summary=TRUE)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: zo, apr 23, 2017 - 14:35:36

db1N == 0, NA, db1$N) stargazer(select(db1, yesN, N, N0) %>% data.frame, summary=TRUE)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: zo, apr 23, 2017 - 14:35:36

# ANALYSIS

# run the model

CD <- sfa(logyld ~ logN + loglab + logasset + + logarea + herb + + mech + elevation + manure + + SOC2 + phdum2 + crop\_count2 + yesN, data=db1)

out <- round(summary(CD)$mleParam %>% data.frame, 3) names(out) <- c("estimate", "SE", "Z", "p") stargazer(out, summary = FALSE)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: zo, apr 23, 2017 - 14:35:40

TL <- sfa(logyld ~ logN + loglab + logasset + + logNsq + loglabsq + logassetsq + + logNlab + logNasset + loglabasset + + logarea + herb + manure + + mech + elevation + + SOC2 + phdum2 + crop\_count2 + yesN, data=db1)

lrtest(TL, CD) # TL model fits better Likelihood ratio test

Model 1: TL Model 2: CD #Df LogLik Df Chisq Pr(>Chisq)  
1 21 -1721  
2 15 -1732 -6 21.4 0.0015  **--- Signif. codes: 0 ‘**\*’ 0.001 ‘\*\*’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 > > # table of results > sf\_tab <- round(as.data.frame(summary(CD)$mleParam), 3) > saveRDS(sf\_tab, "Cache/CD\_sf\_res.rds") > > # we want to evaluate the frontier function > # and avoid the Z variables > xvars <- names(coef(TL))[1:(length(coef(TL))-2)] > X <- TL$dataTable[, xvars] > X <- as.data.frame(X) > xcoef <- coef(TL)[1:(length(coef(TL))-2)] > relprices <- db1validObs] > > # function to calculate the MPP > calc\_mpp <- function(N, row){ + row$logN <- log(N) # add one because we cannot take log(0) + logY <- as.matrix(row) %\*% xcoef + Y <- exp(logY) + MPP <- with(row, ((xcoef["logN"] + + 2\*xcoef["logNsq"]\*logN + + xcoef["logNlab"]\*loglab + + xcoef["logNasset"]\*logasset )\* + (Y/N))) + MPP + } > > # function to calculate the mpp less > # the relative price > mpp\_price <- function(N, row, relprice){ + calc\_mpp(N, row) - relprice + } > > # function to calculate the economically > # optimal level of nitrogen > econ\_opt <- function(i){ + + # get a plot observation and relative + # price + row <- X[i, ] + relprice <- relprices[i] + + # first we need a point above zero + # but before the root to act as the + # lower limit of the root finding + # function - We cap at 700 because + # even on the most productive farms + # 700 kg/ha of nitrogen is excessive + lower <- tryCatch(optimize(function(x) mpp\_price(x, row, relprice), + interval=c(1, 700), + maximum =TRUE)$maximum, error=function(err) NA) +*  
*+ # then we need to find the root + # i.e. the point at which the + # nitrogen level crosses zero. + # this is the economically + # constrained level of nitrogen + tryCatch(uniroot(function(x) {mpp\_price(x, row, relprice)}, + interval=c(lower, 100000))$root, + error=function(err) NA) + } > > # calculate the economically > # optimal level of nitrogen per plot > db2 <- db1[TL$validObs, ] > db2$Npm <- sapply(1:nrow(X), econ\_opt) > > # results are not good -> try cobb douglass > # instead > # MPP values make much more sense > db2$mpp <- coef(CD)["logN"]*db1N > db2mpp == Inf] <- NA > > # find optimal N -> values are very large > db2$Npm <- coef(CD)["logN"]\*db1$yld/(relprices) # sort prices later > > model <- CD > > # 1. Technical efficiency yield is found using the > # output of the sfa model > # Observations where Npm cannot be calculated are removed > db2 <- db2 %>% + rename(Y = yld) %>% + mutate( + Ycor = exp(as.numeric(fitted(model))+log(as.numeric(efficiencies(model)))), + err = Ycor-Y, + TEY = exp(as.numeric(fitted(model))), + TE = as.numeric(efficiencies(model)), + resid = as.numeric(resid(model)) + ) > > # 2. Economic yield is found by evaluating the frontier > # function at the economically optimal nitrogen rate (Npm) > # This means we need to swap out the N (logN) variable > # for the Npm variable, BUT we also need a way of doing > # this for the interaction terms that also involve N > model\_vars <- names(CDolsParam)))] > predict\_dat <- db2[, model\_vars] > predict\_dat <- mutate(predict\_dat, + logN = log(db2$Npm)) > > > # now make the prediction predict.sfa > # function is not made to handle NA values > # so to keep order and compare with other yield > # measures we probably want to set NA values to > # zero temporarily > predict\_dat$logN[is.na(predict\_dat$logN)] <- 0 > db2EY[predict\_dat$logN == 0] <- NA > > # 3. PFY: Feasible yield > # evaluate frontier function at N = 150 > # Increase labour and seed rate by 10% > # turn on all dummies. > predict\_dat2 <- mutate(predict\_dat, + logN = log(200), + loglab = loglab + log(1.5), + logasset = logasset + log(1.5), + herb = 1, + mech = 1) > > # make prediction > db2$PFY <- exp(predict.sfa(model, predict\_dat2)) > > # 4. Potential yield > db2$PY <- db2$YW \* 1000 > > # join all the yield measures into a single > # dataframe > # We cap all values at PY because we consider this as an absolute potential and recalculate all gaps. > db2 <- mutate(db2, PFY = ifelse(PY-PFY<0, PY, PFY), + EY = ifelse(PY-EY<0, PY, EY), + TEY = ifelse(PY-TEY<0, PY, TEY), + Ycor = ifelse(PY-Ycor<0, PY, Ycor), + Y = ifelse(PY-Y<0, PY, Y)) > > # Calculate TYG using UY as reference > db2 <- db2 %>% + mutate( + ERROR\_l = Ycor-Y, # Error gap + ERROR\_s = Y/Ycor, # Error gap + TEYG\_l = TEY-Ycor, # Technical efficiency yield gap using Ycor as basis + TEYG\_s = Ycor/TEY, # Technical efficiency yield gap using Ycor as basis + EYG\_l = EY-TEY, # Economic yield gap + EYG\_s = TEY/EY, # Economic yield gap + EUYG\_l = PFY-EY, # Feasible yield gap + EUYG\_s = EY/PFY, # Feasible yield gap + TYG\_l = PY-PFY, # Technology yield gap + TYG\_s = PFY/PY, # Technology yield gap + YG\_l = PY-Y, # Yield gap + YG\_s = Y/PY, # Yield gap + YG\_l\_Ycor = PY-Ycor, # Yield gap with Ycor as reference + YG\_s\_Ycor = Ycor/PY) # Yield gap with Ycor as reference > > # Consistency check of yield gaps. > # ERROR > X\_ERROR\_check <- filter(db2, ERROR\_l<0) # Half of observation has a negative error which is what would be expected > mean(db2$ERROR\_l, na.rm=TRUE) [1] -111.6 > mean(db2$ERROR\_s, na.rm=TRUE) [1] 1.019 > > # TEYG > X\_TEYG\_check <- filter(db2, TEYG\_l<0) # Should be zero > mean(db2$TEYG\_s, na.rm=TRUE) [1] 0.4524 > > # EYG > # A number of plots will have to decrease N use Npm < N. In several cases also plots that do no use N > # will have lower Y when they start using N. This is because there yield can be located above the frontier (based on fertilizer users) because of the positive effect of noN. > # If we believe that these plots are structurally different and do not use fertilizer because of better soils, they will in fact use too much N and have to decrease. > X\_EYG\_check <- filter(db2, EYG\_l<0)  
> mean(db2$EYG\_s) [1] 0.6122 > > # EUYG > # A number of plots have negative EUYG\_l because Npm is > # larger than Npy, the nitrogen that is required to > # achieve Potential yield (Yw). We have corrected this > # so check should be 0. > # CHECK: we stil find negative values. It is possible > # that because of the interaction with labour >N results > # in <y? TO BE CHECKED > X\_EUYG\_check <- filter(db2, EUYG\_l<0) > mean(db2$EUYG\_s) [1] 0.4401 > check <- select(X\_EUYG\_check, hhid, holder\_id, parcel\_id, field\_id, + surveyyear, ZONE, REGNAME, area, crop\_count2, + lat, lon, noN, yesN, loglab, lab, Npm, mpp, Y, + PFY, EY, EUYG\_l) Error in eval(expr, envir, enclos) : object 'holder\_id' not found > > # TYG > X\_TYG\_check <- filter(db2, TYG\_l<0)  
> mean(db2$TYG\_s, na.rm=TRUE) [1] 0.6205 > > #YG > X\_YG\_check <- filter(db2, YG\_l<0)  
> YG\_check2 <- filter(db2, YG\_l\_Ycor<0)  
> > # Check if separate yield gaps add up to total yield gap > Overall\_check <- db2 %>% + mutate(check\_l = YG\_l/(ERROR\_l + TEYG\_l + EYG\_l + EUYG\_l + TYG\_l), # Note that for a small number of observatios YG\_l=0 resulting in 0/0 which is NaN + check\_s = YG\_s/(ERROR\_s \* TEYG\_s \* EYG\_s \* EUYG\_s \* TYG\_s), + check\_l2 = YG\_l\_Ycor/(TEYG\_l + EYG\_l + EUYG\_l + TYG\_l), + check\_s2 = YG\_s\_Ycor/(TEYG\_s \* EYG\_s \* EUYG\_s \* TYG\_s)) > summary(Overall\_check) hhid id ZONE REGNAME plotno  
Length:1226 Min. : 30 Min. :1.00 Length:1226 Min. :1.00  
Class :character 1st Qu.: 455 1st Qu.:1.00 Class :character 1st Qu.:1.00  
Mode :character Median : 820 Median :2.00 Mode :character Median :1.00  
Mean : 796 Mean :1.61 Mean :1.47  
3rd Qu.:1149 3rd Qu.:2.00 3rd Qu.:2.00  
Max. :1553 Max. :3.00 Max. :5.00

AEZ fs SOC SOC2

Tropic - warm / subhumid:900 Artisanal fishing : 83 Min. : 1.14 Min. : 1.53  
Tropic - warm / humid :326 Cereal-root crop mixed :445 1st Qu.: 2.69 1st Qu.: 4.76  
Humid lowland tree crop:350 Median : 4.01 Median : 6.98  
Root and tuber crop :348 Mean : 4.20 Mean : 7.57  
3rd Qu.: 5.33 3rd Qu.: 9.03  
Max. :13.33 Max. :30.77

ph ph2 RootDepth elevation Y crop\_qty\_harv

Min. :50.0 Min. :49.7 Min. : 0.0 Min. : 0 Min. : 1 Min. : 4  
1st Qu.:56.8 1st Qu.:56.2 1st Qu.: 52.9 1st Qu.:138 1st Qu.: 207 1st Qu.: 209  
Median :60.1 Median :60.0 Median : 74.0 Median :194 Median : 492 Median : 419  
Mean :59.4 Mean :59.2 Mean : 74.5 Mean :202 Mean : 718 Mean : 833  
3rd Qu.:61.8 3rd Qu.:61.7 3rd Qu.: 98.4 3rd Qu.:257 3rd Qu.: 920 3rd Qu.: 838  
Max. :68.4 Max. :71.3 Max. :133.1 Max. :507 Max. :8623 Max. :18147

lab\_val1 lab\_val2 lab\_val3 lab\_val4 implmt\_valu lvstk\_valu

Min. : 1.0 Min. : 53 Min. : 8 Min. : 33 Min. : 0 Min. : 0  
1st Qu.: 37.0 1st Qu.:228 1st Qu.: 48 1st Qu.:189 1st Qu.: 12 1st Qu.: 0  
Median : 49.0 Median :326 Median : 63 Median :281 Median : 24 Median : 60  
Mean : 53.8 Mean :308 Mean : 70 Mean :254 Mean : 97 Mean : 553  
3rd Qu.: 64.0 3rd Qu.:371 3rd Qu.: 84 3rd Qu.:321 3rd Qu.: 48 3rd Qu.: 380  
Max. :400.0 Max. :879 Max. :357 Max. :569 Max. :20070 Max. :35850

fung herb insec manure mech

Min. :0.0000 Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.000  
1st Qu.:0.0000 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000  
Median :0.0000 Median :0.000 Median :0.0000 Median :0.0000 Median :0.000  
Mean :0.0024 Mean :0.205 Mean :0.0106 Mean :0.0514 Mean :0.275  
3rd Qu.:0.0000 3rd Qu.:0.000 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.000  
Max. :1.0000 Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.000

N irrig area area\_tot dist\_hh

Min. : 0.00 Min. :0.0000 Min. :0.020 Min. : 0.08 Length:1226  
1st Qu.: 0.00 1st Qu.:0.0000 1st Qu.:0.607 1st Qu.: 1.21 Class :character  
Median : 0.00 Median :0.0000 Median :1.214 Median : 2.36 Mode :character  
Mean : 4.23 Mean :0.0367 Mean :1.445 Mean : 2.92  
3rd Qu.: 0.00 3rd Qu.:0.0000 3rd Qu.:1.779 3rd Qu.: 4.05  
Max. :158.67 Max. :1.0000 Max. :9.010 Max. :40.47

crop\_count rural lat lon YW phdum  
Min. :1.00 Urban: 75 Min. : 5.21 Min. :-2.8103 Min. : 4.55 1: 114  
1st Qu.:1.00 Rural:1151 1st Qu.: 6.55 1st Qu.:-1.6733 1st Qu.: 7.92 2:1112  
Median :2.00 Median : 7.49 Median :-0.6979 Median : 8.10  
Mean :2.42 Mean : 7.94 Mean :-0.8316 Mean : 8.40  
3rd Qu.:3.00 3rd Qu.: 9.56 3rd Qu.:-0.0293 3rd Qu.: 8.41  
Max. :5.00 Max. :11.12 Max. : 0.8809 Max. :11.11

phdum2 crop\_count2 logyld yesN noN

Min. :0.00 Min. :0.00 Min. :-0.045 Min. :0.000 Min. :0.00  
1st Qu.:1.00 1st Qu.:0.00 1st Qu.: 5.333 1st Qu.:0.000 1st Qu.:1.00  
Median :1.00 Median :0.00 Median : 6.199 Median :0.000 Median :1.00  
Mean :0.87 Mean :0.29 Mean : 6.010 Mean :0.171 Mean :0.83  
3rd Qu.:1.00 3rd Qu.:1.00 3rd Qu.: 6.824 3rd Qu.:0.000 3rd Qu.:1.00  
Max. :1.00 Max. :1.00 Max. : 9.089 Max. :1.000 Max. :1.00

logN lab loglab logarea asset

Min. :-2.091 Min. : 58 Min. : 4.07 Min. :-3.888 Min. : 0  
1st Qu.: 0.000 1st Qu.: 371 1st Qu.: 5.92 1st Qu.:-0.499 1st Qu.: 12  
Median : 0.000 Median : 617 Median : 6.42 Median : 0.194 Median : 49  
Mean : 0.473 Mean : 886 Mean : 6.43 Mean : 0.042 Mean : 258  
3rd Qu.: 0.000 3rd Qu.: 985 3rd Qu.: 6.89 3rd Qu.: 0.575 3rd Qu.: 213  
Max. : 5.067 Max. :28763 Max. :10.27 Max. : 2.198 Max. :13408

logasset logNsq loglabsq logassetsq logNlab

Min. :0.00 Min. : 0.00 Min. : 16.6 Min. : 0.00 Min. :-13.6  
1st Qu.:2.59 1st Qu.: 0.00 1st Qu.: 35.0 1st Qu.: 6.72 1st Qu.: 0.0  
Median :3.92 Median : 0.00 Median : 41.3 Median :15.33 Median : 0.0  
Mean :4.00 Mean : 1.51 Mean : 42.0 Mean :19.20 Mean : 3.1  
3rd Qu.:5.36 3rd Qu.: 0.00 3rd Qu.: 47.5 3rd Qu.:28.77 3rd Qu.: 0.0  
Max. :9.50 Max. :25.67 Max. :105.4 Max. :90.32 Max. : 40.2

logNasset loglabasset DISCODE Pm Pn  
Min. :-6.77 Min. : 0.0 Min. : 1.0 Min. :0.0107 Min. :0.583  
1st Qu.: 0.00 1st Qu.:16.2 1st Qu.: 31.0 1st Qu.:0.2203 1st Qu.:3.333  
Median : 0.00 Median :24.9 Median :109.0 Median :0.2866 Median :4.000  
Mean : 2.29 Mean :26.0 Mean : 93.2 Mean :0.3154 Mean :3.991  
3rd Qu.: 0.00 3rd Qu.:35.0 3rd Qu.:142.0 3rd Qu.:0.3582 3rd Qu.:4.133  
Max. :33.97 Max. :78.6 Max. :171.0 Max. :0.6052 Max. :8.959

Pnm Pns relprice N0 Npm

Min. :0.583 Min. :0.333 Min. : 0.93 Min. : 0.1 Min. : 0.1  
1st Qu.:3.833 1st Qu.:1.000 1st Qu.: 2.64 1st Qu.: 9.3 1st Qu.: 10.3  
Median :4.667 Median :1.067 Median : 3.84 Median : 18.5 Median : 24.2  
Mean :4.600 Mean :1.285 Mean : 5.13 Mean : 24.8 Mean : 42.2  
3rd Qu.:5.333 3rd Qu.:1.100 3rd Qu.: 5.35 3rd Qu.: 37.1 3rd Qu.: 49.2  
Max. :9.930 Max. :4.299 Max. :109.98 Max. :158.7 Max. :618.4  
NA's :1017  
mpp Ycor err TEY TE  
Min. : 0.1 Min. : 3 Min. :-7521 Min. : 239 Min. :0.0106  
1st Qu.: 4.9 1st Qu.: 293 1st Qu.: -100 1st Qu.: 794 1st Qu.:0.3146  
Median : 9.7 Median : 506 Median : 49 Median :1105 Median :0.4718  
Mean : 23.7 Mean : 607 Mean : -112 Mean :1329 Mean :0.4524  
3rd Qu.: 15.5 3rd Qu.: 795 3rd Qu.: 108 3rd Qu.:1638 3rd Qu.:0.6002  
Max. :970.4 Max. :4361 Max. : 466 Max. :6022 Max. :0.8482  
NA's :1017  
resid EY PFY PY ERROR\_l  
Min. :-5.844 Min. : 212 Min. : 1305 Min. : 4552 Min. :-7521  
1st Qu.:-1.592 1st Qu.:1331 1st Qu.: 3670 1st Qu.: 7924 1st Qu.: -100  
Median :-0.935 Median :1959 Median : 4846 Median : 8095 Median : 49  
Mean :-1.034 Mean :2344 Mean : 5109 Mean : 8397 Mean : -112  
3rd Qu.:-0.366 3rd Qu.:3018 3rd Qu.: 6556 3rd Qu.: 8411 3rd Qu.: 108  
Max. : 1.892 Max. :9047 Max. :11107 Max. :11107 Max. : 466

ERROR\_s TEYG\_l TEYG\_s EYG\_l EYG\_s

Min. :0.275 Min. : 98 Min. :0.0106 Min. :-1261 Min. :0.262  
1st Qu.:0.647 1st Qu.: 379 1st Qu.:0.3146 1st Qu.: 385 1st Qu.:0.463  
Median :0.832 Median : 590 Median :0.4718 Median : 798 Median :0.560  
Mean :1.019 Mean : 723 Mean :0.4524 Mean : 1014 Mean :0.612  
3rd Qu.:1.155 3rd Qu.: 918 3rd Qu.:0.6002 3rd Qu.: 1363 3rd Qu.:0.710  
Max. :7.822 Max. :3734 Max. :0.8482 Max. : 6276 Max. :2.215

EUYG\_l EUYG\_s TYG\_l TYG\_s YG\_l

Min. :-407 Min. :0.118 Min. : 0 Min. :0.119 Min. : 0  
1st Qu.:1958 1st Qu.:0.328 1st Qu.:1459 1st Qu.:0.439 1st Qu.: 6889  
Median :2634 Median :0.411 Median :3301 Median :0.597 Median : 7668  
Mean :2765 Mean :0.440 Mean :3288 Mean :0.621 Mean : 7679  
3rd Qu.:3484 3rd Qu.:0.520 3rd Qu.:4700 3rd Qu.:0.814 3rd Qu.: 8026  
Max. :7696 Max. :1.085 Max. :9788 Max. :1.000 Max. :11097

YG\_s YG\_l\_Ycor YG\_s\_Ycor check\_l check\_s check\_l2

Min. :0.0001 Min. : 3199 Min. :0.0004 Min. :1 Min. :1 Min. :1  
1st Qu.:0.0249 1st Qu.: 7090 1st Qu.:0.0352 1st Qu.:1 1st Qu.:1 1st Qu.:1  
Median :0.0580 Median : 7612 Median :0.0609 Median :1 Median :1 Median :1  
Mean :0.0879 Mean : 7790 Mean :0.0744 Mean :1 Mean :1 Mean :1  
3rd Qu.:0.1088 3rd Qu.: 7984 3rd Qu.:0.0985 3rd Qu.:1 3rd Qu.:1 3rd Qu.:1  
Max. :1.0000 Max. :11076 Max. :0.5504 Max. :1 Max. :1 Max. :1  
NA's :1  
check\_s2 Min. :1  
1st Qu.:1  
Median :1  
Mean :1  
3rd Qu.:1  
Max. :1

# Create database with relevant variables for further analysis

db3 <- select(db2, hhid, ZONE, + REGNAME, crop\_count2, area, Pns, Pm, mpp, relprice, + Npm, yesN, Y, N, Ycor, PY, TEY, EY, PFY, PY, ERROR\_l, + ERROR\_s, TEYG\_l, TEYG\_s, EYG\_l, EYG\_s, EUYG\_l, + EUYG\_s, TYG\_l, TYG\_s, YG\_l, YG\_s, YG\_l\_Ycor, YG\_s\_Ycor)

# save db3 for further analysis

saveRDS(db3, "Cache/db3.rds") options(scipen=999) # surpress scientific notation options("stringsAsFactors"=FALSE) # ensures that characterdata that is loaded (e.g. csv) is not turned into factors options(digits=4)

### SOURCE

source(file.path(root, "Code/waterfall\_plot.r"))

### LOAD DATA

db3 <- readRDS(file.path(root, "Cache/db3.rds"))

# in the original data, what we call a zone is

# called a region.

db3$REGNAME <- gsub(" REGION", "", db3$REGNAME) db3REGNAME db3mpp > 50|db3mpp) # remove outliers

# basic zonal summary

by\_zone <- group\_by(db3, ZONE) %>% + filter(!ZONE %in% c("GREATER ACCRA", "WESTERN")) %>% + summarise(n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) Total <- summarise(db3, ZONE = "Total", n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) rbind(by\_zone, Total) # A tibble: 9 × 6 ZONE n Yield Yf Ynf PY \* 1 ASHANTI 181 748.3 840.1 741.2 9154 2 BRONG AHAFO 139 671.2 1161.2 652.9 7906 3 CENTRAL 90 823.5 1494.2 775.6 9299 4 EASTERN 150 806.4 1512.3 549.7 9012 5 NORTHERN 346 720.2 984.8 606.5 7804 6 UPPER EAST 40 616.0 688.6 517.8 8006 7 UPPER WEST 100 466.2 710.9 450.6 8103 8 VOLTA 152 786.5 1106.1 763.9 8305 9 Total 1226 718.1 1057.6 648.3 8397 by\_zone <- group\_by(db3, ZONE) %>% + summarise(n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) Total <- summarise(db3, ZONE = "Total", n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) rbind(by\_zone, Total) # A tibble: 11 × 6 ZONE n Yield Yf Ynf PY \* 1 ASHANTI 181 748.3 840.1 741.2 9154 2 BRONG AHAFO 139 671.2 1161.2 652.9 7906 3 CENTRAL 90 823.5 1494.2 775.6 9299 4 EASTERN 150 806.4 1512.3 549.7 9012 5 GREATER ACCRA 17 586.0 641.6 578.6 7924 6 NORTHERN 346 720.2 984.8 606.5 7804 7 UPPER EAST 40 616.0 688.6 517.8 8006 8 UPPER WEST 100 466.2 710.9 450.6 8103 9 VOLTA 152 786.5 1106.1 763.9 8305 10 WESTERN 11 602.0 NaN 602.0 11107 11 Total 1226 718.1 1057.6 648.3 8397 #'======================================================================================================================================== #' Project: IMAGINE #' Subject: Script to create figures #' Author: Michiel van Dijk #' Contact: [michiel.vandijk@wur.nl](mailto:michiel.vandijk@wur.nl) #'========================================================================================================================================

### PACKAGES

if(!require(pacman)) install.packages("pacman") # Key packages p\_load("tidyverse", "readxl", "stringr", "scales", "RColorBrewer", "rprojroot") # Spatial packages #p\_load("rgdal", "ggmap", "raster", "rasterVis", "rgeos", "sp", "mapproj", "maptools", "proj4", "gdalUtils") # Additional packages p\_load("frontier")

### DETERMINE ROOT PATH

root <- "c:/users/morle001/WEcR/GHAYG"

### DATAPATH

### R SETTINGS

options(scipen=999) # surpress scientific notation options("stringsAsFactors"=FALSE) # ensures that characterdata that is loaded (e.g. csv) is not turned into factors options(digits=4)

### SOURCE

source(file.path(root, "Code/waterfall\_plot.r"))

### LOAD DATA

db3 <- readRDS(file.path(root, "Cache/db3.rds"))

# in the original data, what we call a zone is

# called a region.

db3$REGNAME <- gsub(" REGION", "", db3$REGNAME) db3REGNAME db3mpp > 50|db3mpp) # remove outliers

# basic zonal summary of yields

by\_zone <- group\_by(db3, ZONE) %>% + summarise(n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) Total <- summarise(db3, ZONE = "Total", n = n(), + Yield = round(mean(Y), 2), + Yf = round(mean(Y[yesN == 1]), 2), + Ynf = round(mean(Y[yesN == 0]), 2), + PY = round(mean(PY), 2)) rbind(by\_zone, Total) # A tibble: 11 × 6 ZONE n Yield Yf Ynf PY \* 1 ASHANTI 181 748.3 840.1 741.2 9154 2 BRONG AHAFO 139 671.2 1161.2 652.9 7906 3 CENTRAL 90 823.5 1494.2 775.6 9299 4 EASTERN 150 806.4 1512.3 549.7 9012 5 GREATER ACCRA 17 586.0 641.6 578.6 7924 6 NORTHERN 346 720.2 984.8 606.5 7804 7 UPPER EAST 40 616.0 688.6 517.8 8006 8 UPPER WEST 100 466.2 710.9 450.6 8103 9 VOLTA 152 786.5 1106.1 763.9 8305 10 WESTERN 11 602.0 NaN 602.0 11107 11 Total 1226 718.1 1057.6 648.3 8397 stargazer(by\_zone, summary=FALSE, type = "html")

ZONE

n

Yield

Yf

Ynf

PY

1

ASHANTI

181

748.27

840.12

741.16

9153.91

2

BRONG AHAFO

139

671.16

1161.25

652.87

7905.96

3

CENTRAL

90

823.55

1494.17

775.65

9298.56

4

EASTERN

150

806.4

1512.26

549.72

9011.68

5

GREATER ACCRA

17

586.01

641.6

578.6

7923.64

6

NORTHERN

346

720.2

984.77

606.49

7804.28

7

UPPER EAST

40

616.03

688.6

517.83

8006.42

8

UPPER WEST

100

466.24

710.93

450.62

8102.56

9

VOLTA

152

786.45

1106.06

763.94

8304.82

10

WESTERN

11

602.02

NaN

602.02

11106.83

# nitrogen users, prices, mpp and avcr

by\_zone <- group\_by(db3, ZONE) %>% + summarise(n = mean(N, na.rm=TRUE), + n1 = mean(N[yesN == 1], na.rm=TRUE), + pn = mean(Pns, na.rm=TRUE), + pm = mean(Pm, na.rm=TRUE), + relprice = mean(relprice, na.rm=TRUE), + mpp = mean(mpp[mpp < Inf], na.rm=TRUE)) %>% + data.frame() stargazer(by\_zone, summary=FALSE)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: zo, apr 23, 2017 - 14:43:39