Mehrabi, Z. 2023. Likely decline in the number of farms globally by the middle of the century.

Zia Mehrabi

Feb 14th 2023

Contents

1 Reproducibility

We use the **R** packages knitr (Xie 2022) and checkpoint (Ooi, de Vries, and Microsoft 2022). The package knitr facilitates producing a dynamic document that contains all the steps required to analyze the data. `checkpoint()` will install all packages versions that we used in our analysis to avoid result discrepancies that may arise from software differences. Thus the reader is provided with all the code to fully reproduce the analysis, and adapt it for other analyses – and those doing so can work with and adapt the associated markdown file from which this document was compiled.

```
checkpoint(snapshotDate = "2022-12-22")
```

The document will also be using the dplyr (Wickham, François, et al. 2022), tidyr (Wickham and Girlich 2022), ggplot2 (Wickham, Chang, et al. 2022), FAOSTAT (Kao, Gesmann, and Gheri 2022), qgam (Fasiolo 2021), mgcViz (Fasiolo and Nedellec 2021), ggpubr (Kassambara 2022), ggthemes (Arnold 2021), magick (Ooms 2021), ggokabeito (Barrett 2021), formatR (Xie 2023), readxl (Wickham and Bryan 2022), countrycode (Arel-Bundock 2022), zoo (Zeileis, Grothendieck, and Ryan 2023), kableExtra (Zhu 2021), gridExtra (Auguie 2017) packages.

2 Aim of the document

The aim of this document is describe the data development and analysis underlying Mehrabi et al. (Mehrabi 2023a).Please refer to the published paper for details (or the pre-print located here: (Mehrabi 2023b)), the DOI for the code and the data redirects to this repository. This document includes code for both data development and the analysis, including figure production for the paper.

3 Data

For simplicity all required data are stored and included with this analysis package in the directory "Data".

3.1 Observed agricultural area, population & GDP

First we combine the FAOSTAT data vectors of interest. This includes agricultural area (1000ha), gross domestic product per capita (2005 US2005) and rural population size (thousands) acquired in the bulk download from the UN FAO statistical database (https://www.fao.org/faostat/en/), covering 211 countries and geographies over the years 1960-2013. There are multiple steps outlined below, which read the data, extract variables, harmonize by geographic coverage and clean it.

```
# read dat
pop <- read.csv("Data/FAOSTAT/Population_E_All_Data_(Norm).csv")
macro <- read.csv("Data/FAOSTAT/Macro-Statistics_Key_Indicators_E_All_Data_(Norm).csv")
land <- read.csv("Data/FAOSTAT/Inputs_Land_E_All_Data_(Norm).csv")
land["Year"][land["Year"] == 1961] <- 1960
pc <- translateCountryCode(pop,
  from = "FAOST_CODE", to = "ISO3_CODE",
  "Country.Code")

# subset by vars
land <- subset(land, Item == "Agricultural area" &
  Element == "Area")
```

```
pop.rural <- subset(pop, Item ==
  "Population - Est. & Proj." &
  Element == "Rural population")
pop <- subset(pop, Item == "Population - Est. & Proj." &
  Element == "Total Population - Both sexes")
gdp <- subset(macro, Item == "Gross Domestic Product per capita" &
  Element == "Value US$, 2005 prices")
```

We then add in world bank GDP for 1960 from the World Development Indicators (https://databank.worldbank.org/source/world-development-indicators) which is missing from the FAOSTAT release. Note, the world bank meta-data for these data see screenshot in folder, is wrong, these are 2005 constant US$, to match up to the FAOSTAT release used.

```
gdpwb <- read.csv("Data/WORLDBANK/0fd524e2-575e-4318-badb-0356201384f1_Data.csv",
  skip = 0)
gdpwb <- subset(gdpwb, !(X1960..YR1960. %in%
  c("..", "")))
gdpwb <- translateCountryCode(gdpwb,
  from = "ISO3_WB_CODE", to = "FAOST_CODE",
  "Country.Code")
gdpwb <- subset(gdpwb, !is.na(FAOST_CODE))
gdpwb <- data.frame(Country.Code = gdpwb$FAOST_CODE,
  Country = NA, Item.Code = NA,
  Item = "Gross Domestic Product per capita",
  Element.Code = NA, Element = "Value US$, 2005 prices",
  Year.Code = NA, Year = 1960,
  Unit = NA, Value = gdpwb$X1960..YR1960.,
  Flag = NA)
gdp <- rbind(gdp, gdpwb)
gdp$Value <- as.numeric(gdp$Value)
```

The FAO gdp data does not follow the same geographic boundaries as the land and the pop data (e.g. Serbia and Montenegro for example exist in the latter two datasets but not in the former). Therefore additional steps are required to harmonize these variables. For example I recode Serbia and Montenegro to be one country in gdp data and do the same for Belgium and Luxembourg.

```
gdp$Country.Code<-ifelse(gdp$Country.Code %in% c(272, 273) & gdp$Year %in% c(1990:2005), #SM fix
                         186, gdp$Country.Code)

gdp$Country.Code<-ifelse(gdp$Country.Code %in% c(255, 256) & gdp$Year %in% c(1961:1999), #BL fix
                         15, gdp$Country.Code)


gdp <- gdp %>% group_by(Country.Code, Year) %>%  #aggregate
  summarize(Value=sum(Value, na.rm = T))

#new named vars incl units.
land$Agarea.kha<-land$Value
pop$pop.k<-pop$Value
gdp$gdpcap.US2005<-gdp$Value
pop.rural$pop.rural.k<-pop.rural$Value

#join
```

```
all.fa0<- left_join(gdp,land,  by=c("Country.Code", "Year")) %>%
           left_join(., pop, by=c("Country.Code", "Year")) %>%
            left_join(., pop.rural, by=c("Country.Code", "Year"))

#trim to vars needed
all.fa0<-all.fa0 %>%
  select(Country.Code, Country.x, Year, Agarea.kha,pop.k, gdpcap.US2005,
         pop.rural.k) %>%
  as.data.frame()
```

Further cleaning is neccesary, China is double counted in the FAOSTAT data.

```
# rmv chin
all.fa0 <- subset(all.fa0, !(Country.x %in%
  c("China, Hong Kong SAR", "China, Macao SAR",
    "China, mainland", "China, Taiwan Province of")))
```

We also assign a common code for matching geographies with other non FAO data. And clean up cases where code matching fails or is a many-one.

```
# assign ISO3
all.fa0 <- translateCountryCode(all.fa0,
  from = "FAOST_CODE", to = "ISO3_CODE",
  "Country.Code")

all.fa0$ISO3_CODE <- ifelse(all.fa0$FAOST_CODE ==
  351, "CHN", ifelse(all.fa0$FAOST_CODE ==
  62, "ET", ifelse(all.fa0$FAOST_CODE ==
  277, "SSD", ifelse(all.fa0$FAOST_CODE %in%
  c(247, 246), "YEM", ifelse(all.fa0$FAOST_CODE ==
  15, "BLX", all.fa0$ISO3_CODE)))))

# deal with many-one
all.fa0.t <- all.fa0 %>%
  group_by(ISO3_CODE, Year) %>%
  summarise(Agarea.kha = sum(Agarea.kha),
    pop.k = sum(pop.k), gdpcap.US2005 = sum(gdpcap.US2005),
    isnagdp = sum(is.na(gdpcap.US2005)),
    pop.rural.k = sum(pop.rural.k))

# check gdp data coverage,
# should be full.
summary(all.fa0.t$isnagdp)
all.fa0.t <- all.fa0.t %>%
  select(-isnagdp)
```

There are some minor geographies with missing data for population and agricultural area. One issue in these data in that there is a gap for some geographies with major farming during the soviet union break up over 1990-1991. Here we will in those gaps with the minimum values for each of these in years surrounding the date.

```
# Soviet fix
sov.fix <- subset(all.fao.t, ISO3_CODE %in%
  c("ETH", "ERI", "EST", "UZB",
    "TKM", "TJK", "MDA", "KGZ",
    "GEO", "AZE", "ARM", "UKR",
    "RUS", "LTU", "LVA", "KAZ",
    "BLR", "SVK", "CZE", "CSK",
    "MKD", "SVN", "BIH", "HRV",
    "SCG", "YUG", "SRB", "MNE") &
  Year %in% c(1990:1994))


min.value <- sov.fix %>%
  group_by(ISO3_CODE) %>%
  summarize(Agarea.kha = min(Agarea.kha,
    na.rm = T), pop.k = min(pop.k,
    na.rm = T), gdpcap.US2005 = min(gdpcap.US2005,
    na.rm = T), pop.rural.k = min(pop.rural.k,
    na.rm = T))


tofill <- subset(sov.fix, complete.cases(sov.fix) ==
  F)
tofill <- tofill %>%
  select(ISO3_CODE, Year)

sov.fix <- left_join(tofill, min.value,
  by = "ISO3_CODE")

# and for Sudan
sudan.fix <- subset(all.fao.t,
  ISO3_CODE %in% c("SDN", "SSD") &
    Year %in% c(2008:2011))

min.value <- sudan.fix %>%
  group_by(ISO3_CODE) %>%
  summarize(Agarea.kha = min(Agarea.kha,
    na.rm = T), pop.k = min(pop.k,
    na.rm = T), gdpcap.US2005 = min(gdpcap.US2005,
    na.rm = T), pop.rural.k = min(pop.rural.k,
    na.rm = T))

tofill <- subset(sudan.fix, complete.cases(sudan.fix) ==
  F)
tofill <- tofill %>%
  select(ISO3_CODE, Year, gdpcap.US2005)

sudan.fix <- left_join(tofill,
  min.value, by = "ISO3_CODE")
sudan.fix <- subset(sudan.fix,
  select = c(1, 2, 5, 4, 3, 7))
names(sudan.fix)[5] <- c("gdpcap.US2005")
```

```r
# join to FAO data
all.fa0.t <- rbind(all.fa0.t, sudan.fix,
  sov.fix)

# clean NAs
all.fa0.t <- all.fa0.t[which(!is.na(all.fa0.t$Agarea.kha) |
  !is.na(all.fa0.t$pop.k) | !is.na(all.fa0.t$gdpcap.US2005) |
  !is.na(all.fa0.t$pop.rural.k)),
  ]
```

3.2 Projected population & GDP

Next we add in the SSP data on projected GDP per capita (2005 $US, billions) and rural population size (millions) downloaded in bulk from IIASA's Shared-Socioeconomic Pathways database (https://tntcat.iiasa.ac.at/SspDb), for SSP2, SSP1 and SSP3, covering 182 countries over the years 2015-2100. SSP2 was selected as it represents the 'middle of the road' scenario, whereas SS1 and SSP3 were selected as they represent the most extreme population and GDP per capita scenarios. I convert the units to match those in the FAO data, and only use the projections for 2015-2100. I also correct for the SSP values to match the observed initial conditions in 2013, assuming compounding growth. The long winded function below does all this in one swoop.

```r
getFAOssp <- function(sspx) {

  # grab the future
  # projections
  ssp <- read.csv("Data/IIASA/SspDb_country_data_2013-06-12.csv")

  ssp.pop <- subset(ssp, VARIABLE ==
    "Population" & grepl(sspx,
    ssp$SCENARIO))
  ssp.pop <- pivot_longer(ssp.pop,
    cols = X1950:X2150, names_to = "Year",
    names_prefix = "X", values_to = "Pop.m") %>%
    group_by(REGION, Year) %>%
    summarize(Pop.m = mean(Pop.m,
      na.rm = T))

  ssp.pop.urbshare <- subset(ssp,
    VARIABLE == "Population|Urban|Share" &
      grepl(sspx, ssp$SCENARIO))

  ssp.pop.urbshare <- pivot_longer(ssp.pop.urbshare,
    cols = X2010:X2100, names_to = "Year",
    names_prefix = "X", values_to = "Pop.urb.share") %>%
    group_by(REGION, Year) %>%
    summarize(Pop.urb.share = mean(Pop.urb.share,
      na.rm = T))

  ssp.pop.urbshare <- ssp.pop.urbshare %>%
    group_by(REGION) %>%
    arrange(REGION, Year) %>%
    mutate(pop.urb.i = zoo::na.approx(Pop.urb.share,
      maxgap = 1)) %>%
    select(-Pop.urb.share)

  ssp.gdp <- subset(ssp, VARIABLE ==
    "GDP|PPP" & UNIT == "billion US$2005/yr" &
    grepl(sspx, ssp$SCENARIO))

  ssp.gdp <- pivot_longer(ssp.gdp,
    cols = X1950:X2150, names_to = "Year",
    names_prefix = "X", values_to = "Gdp.US2005.b") %>%
    group_by(REGION, Year) %>%
    summarize(Gdp.US2005.b = mean(Gdp.US2005.b,
      na.rm = T))

  ssp.sub <- left_join(ssp.gdp,
    ssp.pop, by = c("REGION",
      "Year")) %>%
    left_join(., ssp.pop.urbshare,
      by = c("REGION", "Year"))

  # match units to FAOSTAT
  ssp.sub$gdpcap.US2005 <- (ssp.sub$Gdp.US2005.b/ssp.sub$Pop.m) *
    1000
  ssp.sub <- subset(ssp.sub,
    Year > 2010 & Year < 2101)  #get date range
  ssp.sub$pop.k <- ssp.sub$Pop.m *
    1000
  ssp.sub$pop.rural.k <- (100 -
    ssp.sub$pop.urb.i) * ssp.sub$pop.k *
    0.01
  ssp.sub <- na.omit(ssp.sub)

  ssp.sub$REGION <- ifelse(ssp.sub$REGION %in%
    c("TWN", "MAC", "HKG",
      "CHN"), "CHN", as.character(ssp.sub$REGION))

  ssp.sub <- ssp.sub %>%
    group_by(REGION, Year) %>%
    summarize(pop.k = sum(pop.k,
      na.rm = T), gdpcap.US2005 = sum(gdpcap.US2005,
      na.rm = T), pop.rural.k = sum(pop.rural.k,
      na.rm = T))

  max.Area <- all.fa0.t %>%
    filter(Year == 2013) %>%
    select(ISO3_CODE, Agarea.kha)

  # adjust the SSP data to
  # observed record
  gdpFAO <- all.fa0.t %>%
    filter(Year == 2013) %>%
    select(ISO3_CODE, gdpcap.US2005,
      pop.k, pop.rural.k) %>%
    as.data.frame()
  gdpssp <- ssp.sub %>%
    filter(Year == 2015) %>%
    select(REGION, gdpcap.US2005,
      pop.k, pop.rural.k) %>%
    as.data.frame()

  diff <- left_join(gdpssp, gdpFAO,
    by = c(REGION = "ISO3_CODE"))

  diff$diffg <- diff$gdpcap.US2005.y/diff$gdpcap.US2005.x  #FAO = y, ssp = x
  diff$diffp <- diff$pop.k.y/diff$pop.k.x
  diff$diffp.r <- diff$pop.rural.k.y/diff$pop.rural.k.x
  diff <- select(diff, c(REGION,
    diffg, diffp, diffp.r))

  ssp.sub <- left_join(ssp.sub,
    diff, by = c(REGION = "REGION"))
  ssp.sub$gdpcap.US2005 <- ssp.sub$gdpcap.US2005 *
    ssp.sub$diffg
  ssp.sub$pop.k <- ssp.sub$pop.k *
    ssp.sub$diffp
  ssp.sub$pop.rural.k <- ssp.sub$pop.rural.k *
    ssp.sub$diffp.r
  ssp.sub <- select(ssp.sub,
    c(-diffg, -diffp, -diffp.r))

  ssp.sub <- left_join(ssp.sub,
    max.Area, by = c(REGION = "ISO3_CODE"))
  ssp.sub <- subset(ssp.sub,
    select = c(1, 2, 6, 3,
      4, 5))
  names(ssp.sub)[names(ssp.sub) ==
    "REGION"] <- "ISO3_CODE"
  ssp.sub$Year <- as.integer(ssp.sub$Year)

  # join to FAO
  full.dat <- rbind(all.fa0.t,
    ssp.sub)
  return(full.dat)
}
```

Finally we run the function for joining future projections of gdp, population size with the historical observed data on gdp, population size, and agricultural area across SSPs 1-3, representing middle of the road, as well as extreme combinations of high/low gdp, and population size.

```
sspall <- c("SSP1", "SSP2", "SSP3",
  "SSP4")
full.dat <- getFAOssp(sspall[2])  #SSP2, middle road
full.dat1 <- getFAOssp(sspall[1])  #SSP1, high GDP, low pop
full.dat3 <- getFAOssp(sspall[3])  #SSP3, low GDP, high pop

ssp <- full.dat[which(full.dat$Year >
  2013), ]
```

3.3 Observed farm numbers

Next we combine two sources of farm count data. This is taken from two sources - the first is data on the number of farms for different countries and years downloaded from the The World Programme for the Census of Agriculture (https://www.fao.org/world-census-agriculture/en/), and the second data from the supplementary information of publications synthesizing this work for the 2000 and 2010 rounds (Lowder, Sánchez, and Bertini 2021). These data represent a total of 512 national agricultural censuses, covering 175 countries across the years 1960-2015.

For individual census dates spanning multiple years we take the earlier date (in many cases these span only two years), and for any dates for the 2010 census falling 2014/2015 we set the date to 2013 to match observed predictor dataframe end date, there are very few cases in this latter category. These small errors won't be important to parametizing the overall macro-trends (technically all WCA data is centered anyway to the nearest decade).

```
wca <- readxl::read_excel("Data/WCA/Final_1.2_Trends_in_number_and_area_of_holdings_since_1930_WCA_round
  skip = 3)

wca <- wca %>%
  filter(`Data item` == "Number of holdings") %>%
  select(-`Data item`) %>%
  pivot_longer(!`Countries by region`,
    names_to = "Year", values_to = "count")

wca$Year <- as.numeric(wca$Year)
wca$count <- as.numeric(wca$count)

Lowd <- readxl::read_excel("Data/LOWDER2021/1-s2.0-S0305750X2100067X-mmc3.xlsx",
  skip = 1)

Lowd$`Census year` <- as.numeric(substr(Lowd$`Census year`,
  1, 4))

Lowd <- subset(Lowd, `Census year` >
  2005)  #to get data not falling in the WCA

Lowd <- Lowd %>%
  select(Country, `Census year`,
    `Total number of farms`) %>%
  rename(`Countries by region` = Country,
    Year = "Census year", count = "Total number of farms")

Lowd$Year <- ifelse(Lowd$Year >
  2013, 2013, Lowd$Year)

counts <- rbind(wca, Lowd)
counts <- na.omit(counts)
counts <- subset(counts, Year >
  1959)
```

Next we clean up the geographic labels for the holdings data. We note that in some cases, e.g. Slovenia it is not possible to match to the FAO or SSP data.

```r
counts <- as.data.frame(counts)
counts$`Countries by region` <- car::recode(counts$`Countries by region`,
  "c('Venezuela (Bolivarian Republic of)','Venezuela, Bolivarian Republic of') = 'Venezuela';
 c('United States Virgin Islands','Virgin Islands, United States') = 'United States Virgin Islands';
  else=as.character(counts$`Countries by region`)")

counts$ISO3_CODE <- countrycode::countrycode(counts$`Countries by region`,
  origin = "country.name", destination = "iso3c",
  custom_match = c(`Czechoslovakia (former)` = "CSK",
    `Yugoslavia (former)` = "YUG"))

# clean special cases of IS)
# to get max match
counts$ISO3_CODE <- ifelse(counts$ISO3_CODE ==
  "ETH" & counts$Year < 1990,
  "ET", ifelse(counts$ISO3_CODE %in%
    c("BEL", "LUX") & counts$Year <
    1999, "BLX", ifelse(counts$ISO3_CODE ==
    "SRB" & counts$Year ==
    2000, "SCG", counts$ISO3_CODE)))

counts$Year <- ifelse(counts$ISO3_CODE ==
  "CSK" & counts$Year == 1990,
  1989, counts$Year)

counts <- counts %>%
  group_by(ISO3_CODE, Year) %>%
  summarize(count = sum(as.numeric(count)))
```

Finally we join this farm count data to the gdp, population and agricultural area data. There are 40 missing points for 11 minor territories. Many are small islands.

```r
full.dat.ssp2 <- left_join(full.dat,
  counts, by = c("ISO3_CODE",
    "Year"))

full.dat.ssp3 <- left_join(full.dat3,
  counts, by = c("ISO3_CODE",
    "Year"))

full.dat.ssp1 <- left_join(full.dat1,
  counts, by = c("ISO3_CODE",
    "Year"))

pre <- full.dat.ssp2[which(full.dat.ssp2$Year <
  2013), ]
post <- full.dat.ssp2[which(full.dat.ssp2$Year >
  2013), ]

length(unique(pre$ISO3_CODE))
length(unique(post$ISO3_CODE))
```

4 Model fitting

4.1 Overview

I use hierarchical quantile additive regression model to model the association between log (farm number) and the following predictors log(GDP.capita), log (rural population size) and log(agricultural area). The model choice was chosen for its flexibility, i.e. the ability to model parametric and nonparametric, and random regression coefficients, because quantile estimation makes few distributional assumptions, and estimation is invariant the transforms I used to simplify evaluation of the model.

The model I consider for the main analysis is a smoothing term for each parameter and a country random effect, which allows for modeling the correlation of errors (i.e. non-independence) for data collected from the same country over multiple years, with the additional benefit of making adjustments for nations that deviate from the average (i.e. China and Russia). I allow for shrinkage of the smooths to zero.

I compare this model to alternatives, most notably the same model but with no country random effects, a simple linear model with no country random effects; and finally a model with interactions with GDP, as requested by a reviewer who was curious whether that fitted better to the data.

4.2 Pre-processing

The data requires some pre-processing, namely to transform the predictors outside the model formula, remove NAs.

```r
full.dat.ssp2 <- subset(full.dat.ssp2,
  !ISO3_CODE %in% "SGP")  #rmv zero rural pop
full.dat.ssp2$log.gdp <- log(full.dat.ssp2$gdpcap.US2005)
full.dat.ssp2$log.rural.pop <- log(full.dat.ssp2$pop.rural.k)
full.dat.ssp2$log.area <- log(full.dat.ssp2$Agarea.kha)
full.dat.ssp2.na <- na.omit(full.dat.ssp2)
full.dat.ssp2.na$ISO3_CODE <- factor(full.dat.ssp2.na$ISO3_CODE)
```

4.3 Model comparison

Next I fit the main model and the three alternatives.

```r
# simple lm
q1 <- qgamV(log(count) ~ log.area +
  log.gdp + log.rural.pop, qu = 0.5,
  data = full.dat.ssp2.na)

# smooth, re
q2 <- qgamV(log(count) ~ s(log.area,
  bs = "ts") + s(log.gdp, bs = "ts") +
  s(log.rural.pop, bs = "ts") +
  s(ISO3_CODE, bs = "re"), qu = 0.5,
  data = full.dat.ssp2.na)

# smooth, no re
q3 <- qgamV(log(count) ~ s(log.area,
  bs = "ts") + s(log.gdp, bs = "ts") +
  s(log.rural.pop, bs = "ts"),
```

XB	AIC	GCV.UBRE	DEV.EXP
log.area + log.gdp + log.rural.pop	1020.98	518.68	0.77
s(log.area, bs = "ts") + s(log.gdp, bs = "ts") + s(log.rural.pop, bs = "ts") + s(ISO3_CODE, bs = "re")	341.72	422.85	0.94
s(log.area, bs = "ts") + s(log.gdp, bs = "ts") + s(log.rural.pop, bs = "ts")	904.71	475.31	0.81
s(log.area, bs = "ts", by = log.gdp) + s(log.gdp, bs = "ts", by = log.gdp) + s(log.rural.pop, bs = "ts", by = log.gdp) + s(ISO3_CODE, bs = "re")	416.98	456.57	0.94

```
  qu = 0.5, data = full.dat.ssp2.na)

# interactions with gdp
q4 <- qgamV(log(count) ~ s(log.area,
  bs = "ts", by = log.gdp) +
  s(log.gdp, bs = "ts", by = log.gdp) +
  s(log.rural.pop, bs = "ts",
    by = log.gdp) + s(ISO3_CODE,
  bs = "re"), qu = 0.5, data = full.dat.ssp2.na)
```

Then we create a table of the different models considered here, showing the terms and fits. Notably even a really simple linear model explains the variation in the data well.

```
mod <- c(paste((q1$formula)[3]),
  paste((q2$formula)[3]), paste((q3$formula)[3]),
  paste((q4$formula)[3]))
aic <- c(q1$aic, q2$aic, q3$aic,
  q4$aic)
gcv.ubre <- c(q1$gcv.ubre, q2$gcv.ubre,
  q3$gcv.ubre, q4$gcv.ubre)
dev.expl <- c(1 - (q1$deviance/q1$null.deviance),
  1 - (q2$deviance/q2$null.deviance),
  1 - (q3$deviance/q3$null.deviance),
  1 - (q4$deviance/q4$null.deviance))
kableExtra::kbl(data.frame(XB = mod,
  AIC = round(aic, 2), GCV.UBRE = round(gcv.ubre,
    2), DEV.EXP = round(dev.expl,
    2))) %>%
  kableExtra::column_spec(1,
    width = "15em") %>%
  kableExtra::kable_styling()
```

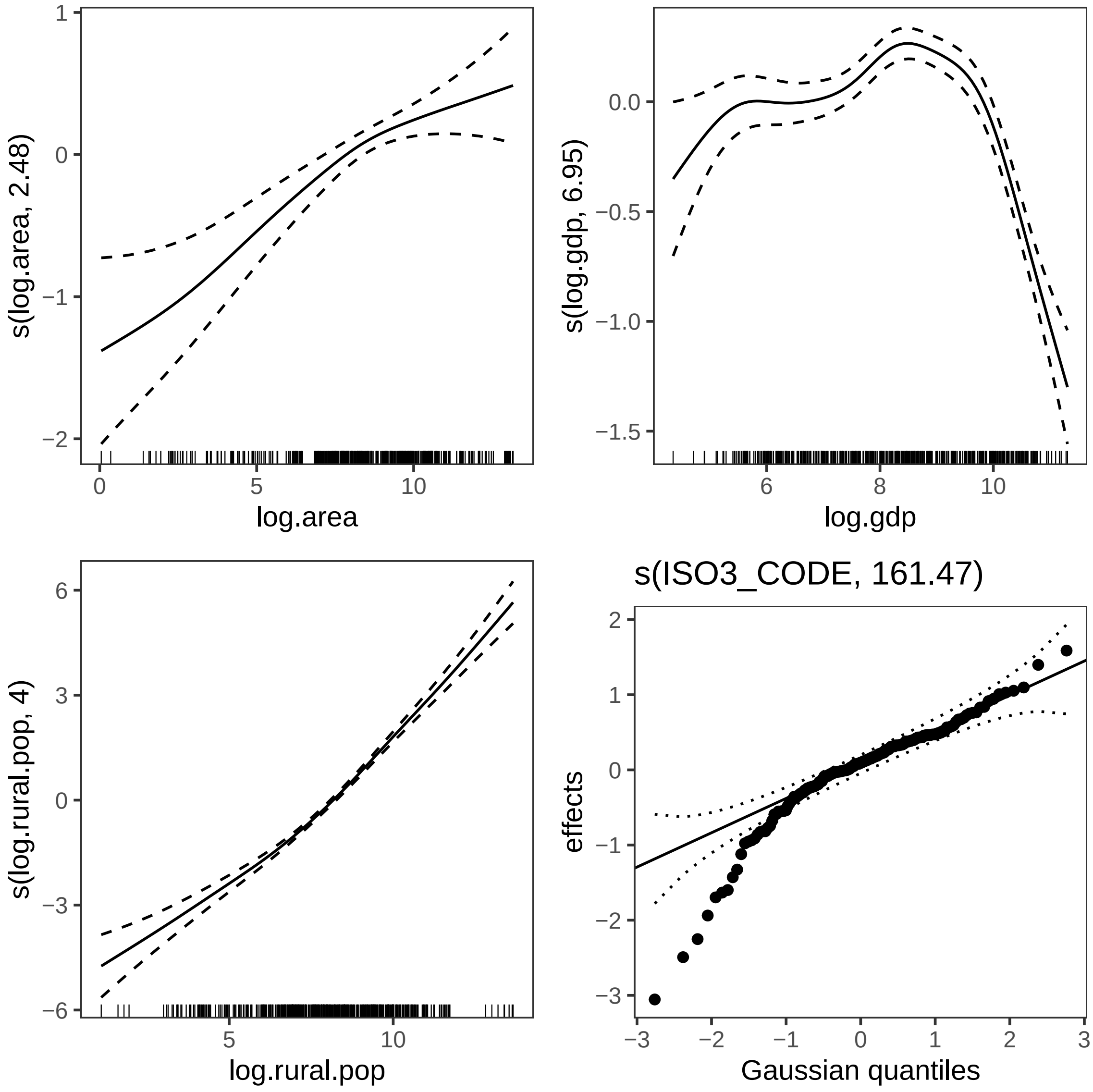
4.4 Model checking

We move with the best fitting model from the set which is the smooth terms with random effects for countries.

4.4.1 Terms

First we check the terms in the model to evaluate each functional form fitted for each.

```
print(plot(q2, allTerms = TRUE),
  pages = 1)
```



```
png(file = "Figs/FigS2.png", width = 12,
  height = 12, units = "cm",
  res = 300)
print(plot(q2, allTerms = TRUE),
  pages = 1)
dev.off()
```

4.4.2 Residual diagnostics

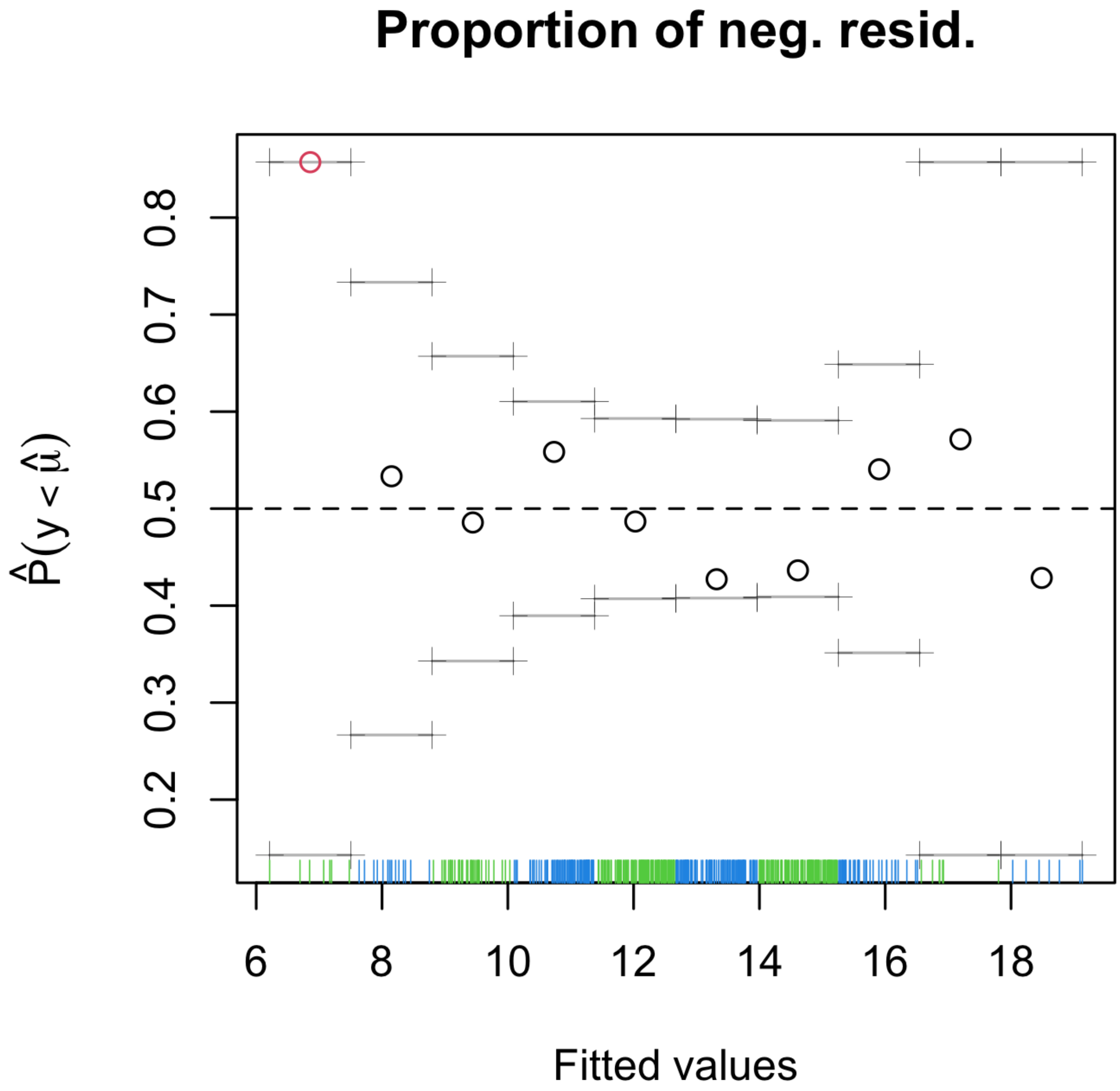
Residual diagnostic plots are created below, illustrated broad validation of the assumptions being made (i.e. of functional form of the predictors and that intended quantiles are being estimated correctly).

```
o <- getViz(q2)
png(file = "Figs/FigS1.png", width = 12,
  height = 12, units = "cm",
  res = 300)
check(o) + l_gridQCheck1D(qu = 0.5)
```

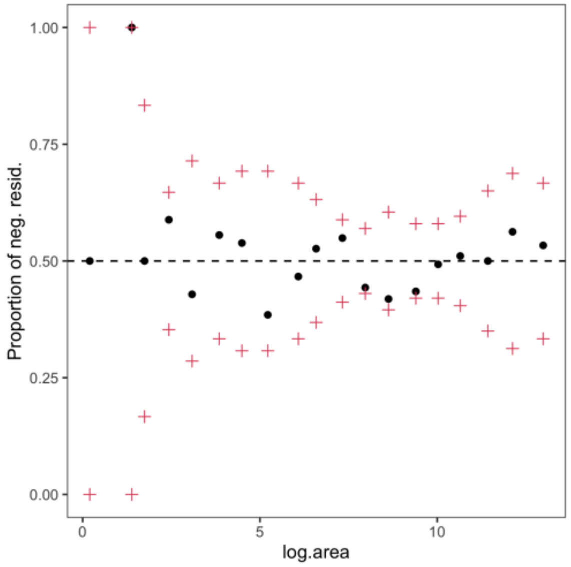
```
## Error in .subset2(x, i, exact = exact): invalid subscript type 'language'
```

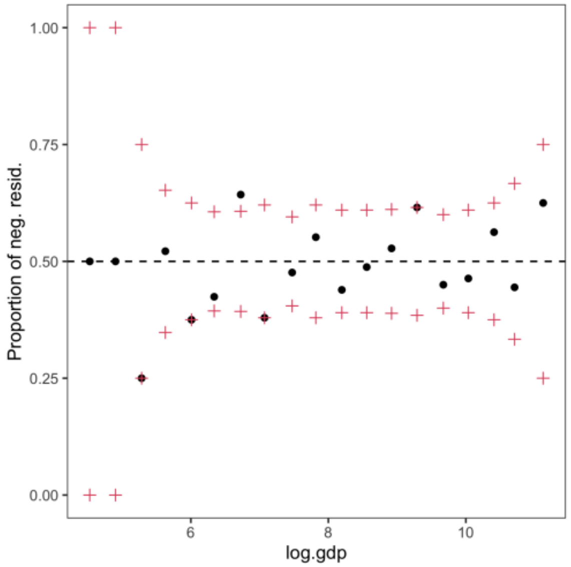
```
dev.off()

l <- list(a = check1D(o, "log.area") +
  l_gridQCheck1D(qu = 0.5), b = check1D(o,
  "log.gdp") + l_gridQCheck1D(qu = 0.5),
  c = check1D(o, "log.rural.pop") +
    l_gridQCheck1D(qu = 0.5))

for (i in 1:length(l)) {
  png(file = paste("Figs/resid",
    i, ".png", sep = ""), width = 12,
    height = 12, units = "cm",
    res = 300)
  print(l[i])
  dev.off()
}
a <- image_read("Figs/resid1.png")
g <- image_read("Figs/resid2.png")
p <- image_read("Figs/resid3.png")
img <- c(a, g, p)
i <- image_append(image_scale(img,
  "x500"), stack = T)
image_write(i, "Figs/FigS3.png")
```

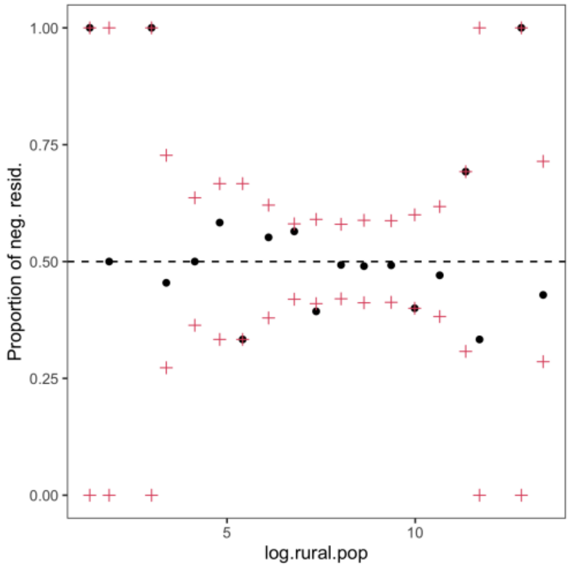
```
modelresid <- image_read("Figs/FigS1.png")
modelresid
```



```
termresid <- image_read("Figs/FigS3.png")
termresid
```


Proportion of neg. resid.
1.00
0.75
0.50
0.25
0.00
0
5
10
log.area

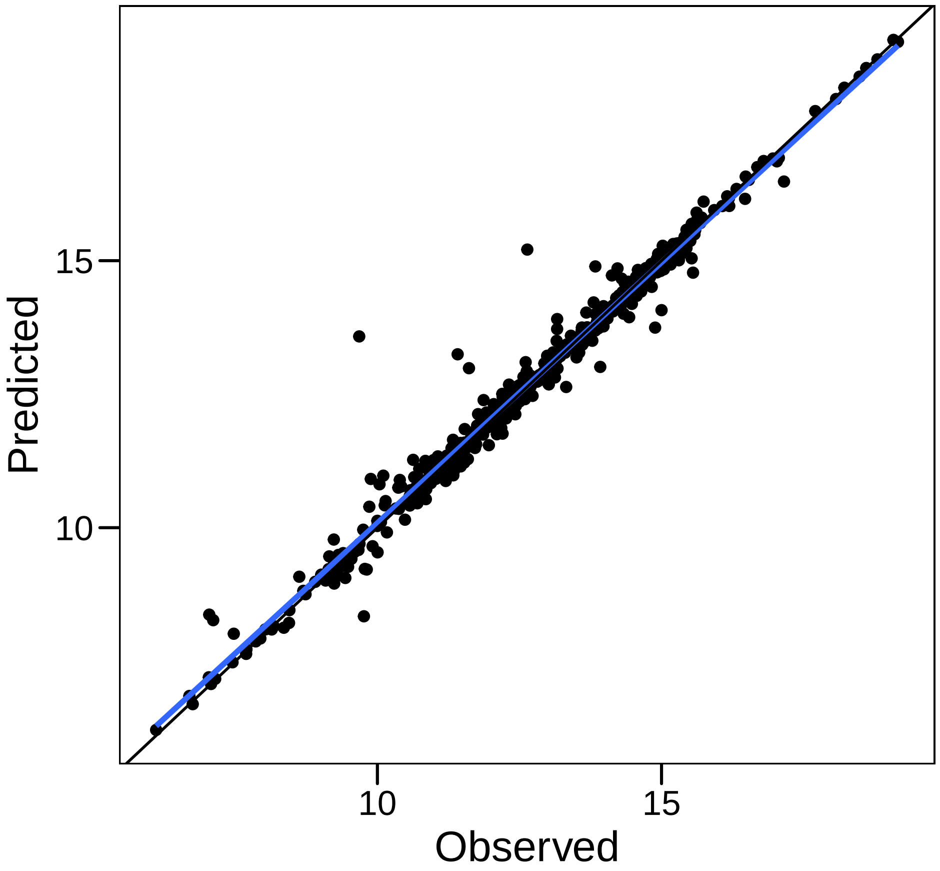

Proportion of neg. resid.
1.00
0.75
0.50
0.25
0.00
6
8
10
log.gdp


Proportion of neg. resid.
1.00
0.75
0.50
0.25
0.00
5
10
log.rural.pop

4.4.3 Predicted versus observed

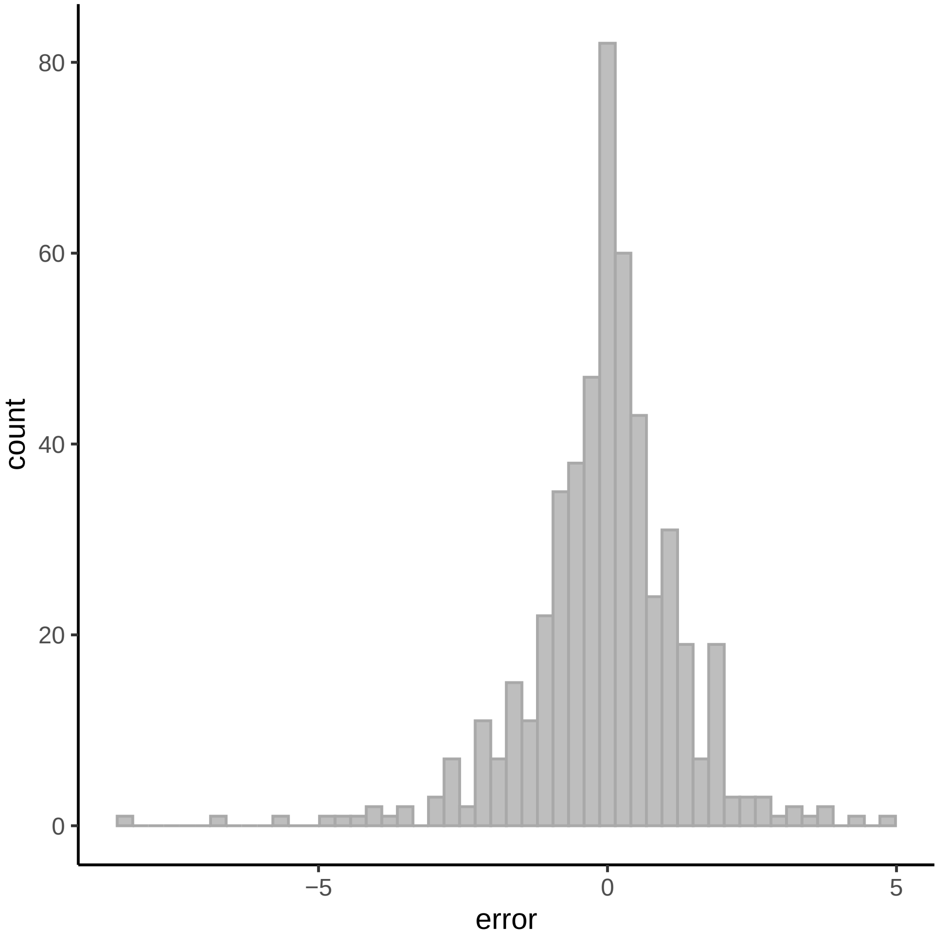
Next we create the predicted versus observed plot for the model. We generate plots on both the log scale and the untransformed response for comparison as shown following sections. Here we just illustrate the log scale results for brevity.

```r
full.dat.ssp2.na$fit <- fitted(q2)
full.dat.ssp2.na$fit.nre <- fitted(q3)

log <- ggplot(full.dat.ssp2.na,
  aes(log(count), (fit))) + geom_point() +
  geom_smooth(method = "lm",
    se = F) + geom_abline(slope = 1,
  intercept = 0) + xlab("Observed") +
  ylab("Predicted") + ggtitle("") +
  theme_base() + theme(plot.background = element_blank())

obser <- ggplot(full.dat.ssp2.na,
  aes(count, exp(fit))) + geom_point() +
  geom_smooth(method = "lm",
    se = F) + geom_abline(slope = 1,
  intercept = 0) + xlab("Observed") +
  ylab("Predicted") + ggtitle("") +
  theme_base() + theme(plot.background = element_blank())

log
```



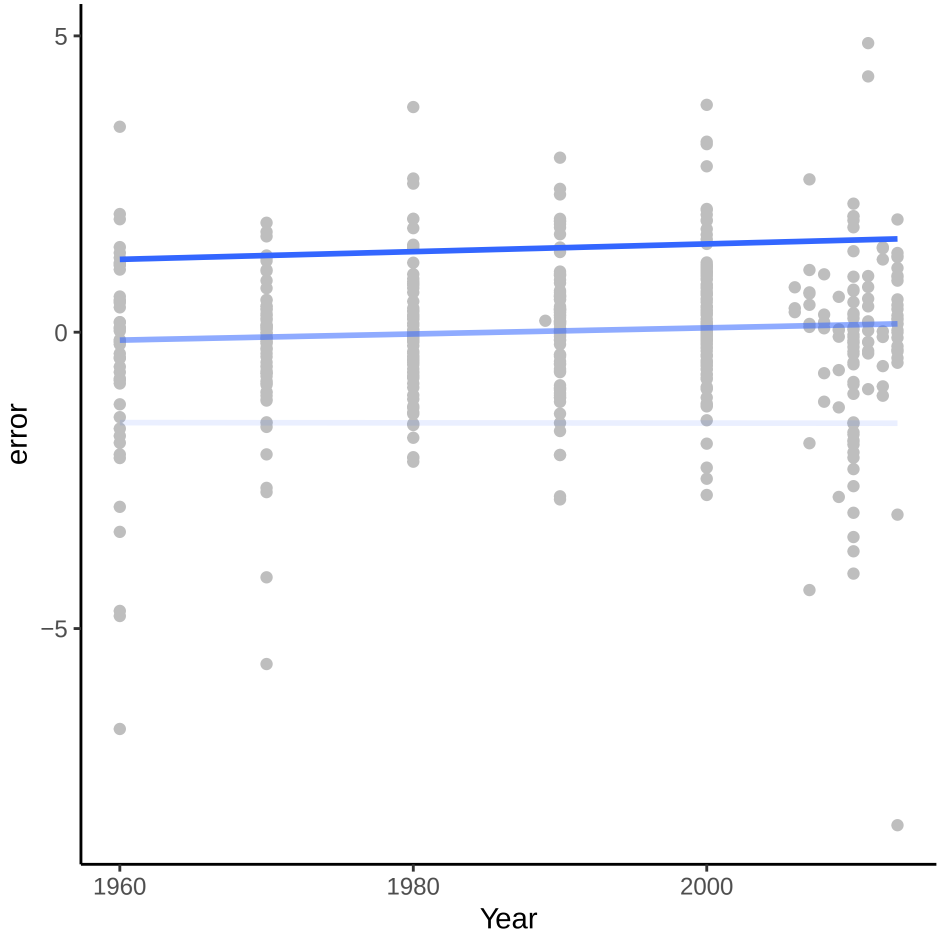
I also create the distribution of errors on each scale.

```
full.dat.ssp2.na$error.obs <- full.dat.ssp2.na$count -
  exp(full.dat.ssp2.na$fit)
full.dat.ssp2.na$error.mod <- residuals(q2)

ero <- ggplot(full.dat.ssp2.na,
  aes(error.obs)) + geom_histogram(colour = "dark gray",
  fill = "gray", bins = 40) +
  theme_classic() + theme(plot.background = element_blank()) +
  scale_x_continuous(breaks = c(-1e+07,
    0, +1e+07)) + xlab("error")

erm <- ggplot(full.dat.ssp2.na,
  aes(error.mod)) + geom_histogram(colour = "dark gray",
  fill = "gray", bins = 50) +
  theme_classic() + theme(plot.background = element_blank()) +
  xlab("error")

erm
```



And the time series of errors with plots of 0.1,0.5 and 0.9 quantiles which indicate no structural temporal effect in the errors as may be expected if there were non-stationarity in the model I was not capturing. Finally add all the plots together and save.

```
geog <- full.dat.ssp2.na %>%
  group_by(ISO3_CODE) %>%
  count() %>%
  filter(n > 0)

to <- full.dat.ssp2.na %>%
  filter(ISO3_CODE %in% geog$ISO3_CODE) %>%
  ggplot(aes(Year, error.obs)) +
  geom_point(colour = "dark gray") +
  geom_smooth(method = "lm",
    se = F) + theme_classic() +
  geom_quantile(quantiles = c(0.1,
    0.5, 0.9), size = 1, aes(alpha = ..quantile..)) +
  theme(plot.background = element_blank()) +
  ylab("error") + theme(legend.position = "none")


tm <- full.dat.ssp2.na %>%
  filter(ISO3_CODE %in% geog$ISO3_CODE) %>%
  ggplot(aes(Year, error.mod)) +
  geom_point(colour = "gray") +
  geom_quantile(quantiles = c(0.1,
    0.5, 0.9), size = 1, aes(alpha = ..quantile..)) +
  # geom_smooth(method='lm',
  # se=F)+
theme_classic() + theme(plot.background = element_blank()) +
  ylab("error") + theme(legend.position = "none")
tm
```



```
a <- obser + annotation_custom(ggplotGrob(ero),
  xmin = -1e+07, xmax = 1e+08,
  ymin = 1e+08, ymax = 2e+08) +
  annotation_custom(ggplotGrob(to),
    xmin = 9e+07, xmax = 2.1e+08,
    ymin = -1e+07, ymax = 9.6e+07) +
  annotate("text", x = -1e+06,
    y = 2e+08, label = "bold(\"e\")",
    parse = TRUE) + annotate("text",
  x = 9.8e+07, y = 8.6e+07, label = "bold(\"f\")",
  parse = TRUE)
```

```
b <- log + annotation_custom(ggplotGrob(erm),
  xmin = 5.5, xmax = 11.5, ymin = 12.5,
  ymax = 19) + annotation_custom(ggplotGrob(tm),
  xmin = 13, xmax = 19, ymin = 5.5,
  ymax = 12) + annotate("text",
  x = 5.9, y = 18.5, label = "bold(\"b\")",
  parse = TRUE) + annotate("text",
  x = 13, y = 11, label = "bold(\"c\")",
  parse = TRUE)

# now save all the data
p <- ggarrange(b, a, labels = c("a",
  "d"), ncol = 2, nrow = 1)
# save
ggsave(file = "Figs/Fig1.png",
  p, dpi = 300, width = 12, height = 5.5)  #saves g
dev.off()
```

```
## null device
##           1
```

4.4.4 Regional breakdowns

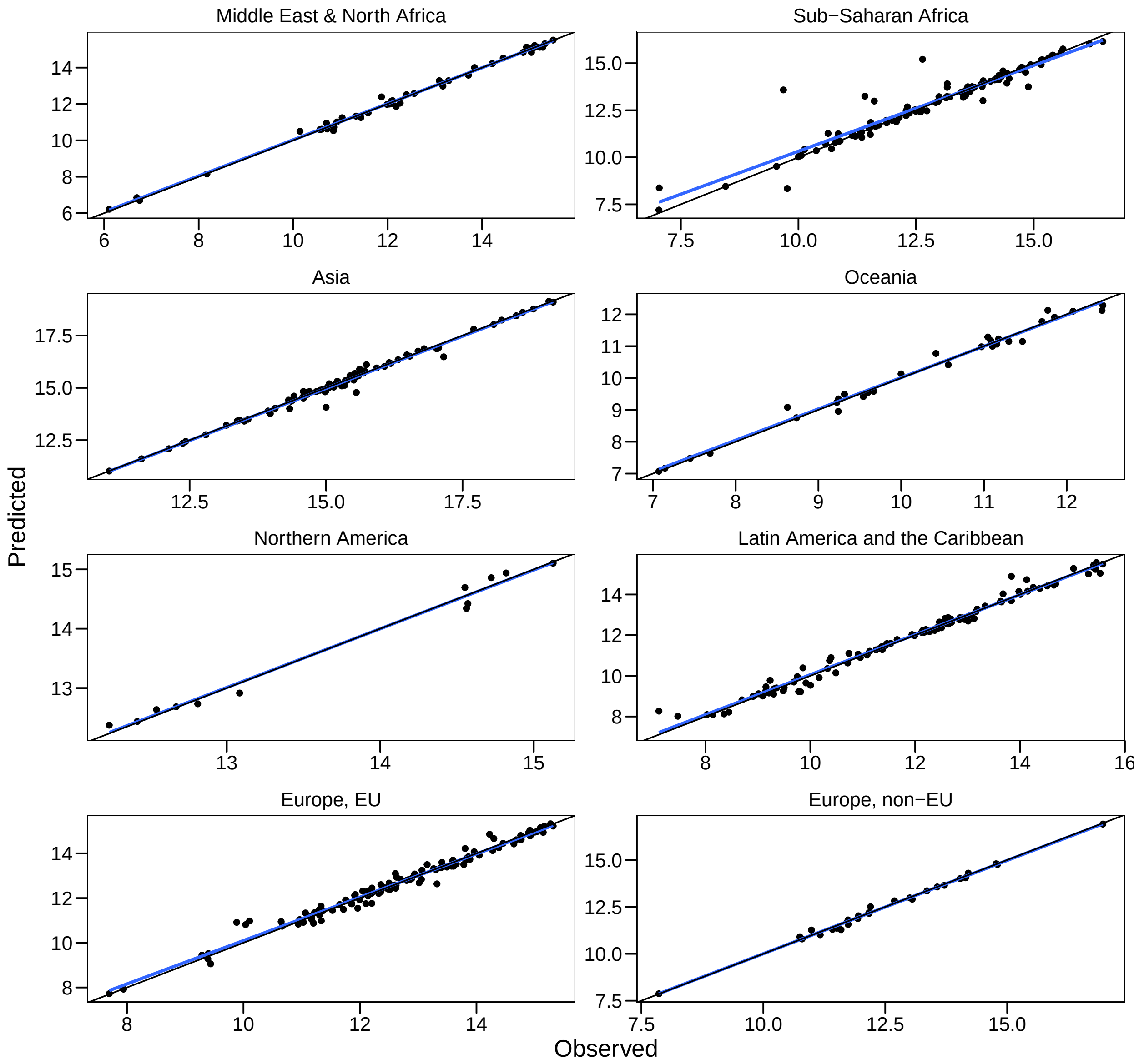
We also check the predictions for regional break downs to see performance on that level, as we build a global model effectively. We first write functions to add regions of interest to the data.

```
add.reg.fun <- function(dat) {
  dat$region <- countrycode::countrycode(dat$ISO3_CODE,
    origin = "iso3c", destination = "un.region.name",
    custom_match = c(CSK = "Europe",
      BLX = "Europe", ET = "Africa",
      SCG = "Europe", SUN = "Europe",
      YUG = "Europe"))
  sub <- countrycode::countrycode(dat$ISO3_CODE,
    origin = "iso3c", destination = "un.regionsub.name",
    custom_match = c(ANT = "Latin America and the Caribbean"))
  dat$region <- ifelse(sub %in%
    c("Latin America and the Caribbean",
      "Northern America"),
    sub, dat$region)
  sub2 <- countrycode::countrycode(dat$ISO3_CODE,
    origin = "iso3c", destination = "eu28")
  sub2 <- ifelse(is.na(sub2) |
    sub2 == "NA", "Europe, non-EU",
    "Europe, EU")
  # sub2[is.na(sub2)|
  # sub2=='NA'] <- 'Europe
  # non-EU'
  dat$region <- ifelse(dat$region %in%
    c("Europe"), sub2, dat$region)

  mena <- c("TUR", "SYR", "YEM",
```

```r
    "SAU", "QAT", "OMN", "LBN",
    "KWT", "JOR", "ISR", "IRN",
    "IRQ", "EGY", "CYP", "BHR",
    "ARE", "DZA", "EGY", "LBY",
    "MAR", "SDN", "TUN")  #no palestine in data.
  dat$region <- ifelse(dat$ISO3_CODE %in%
    mena, "Middle East & North Africa",
    dat$region)
  dat$region <- ifelse(dat$region %in%
    "Africa", "Sub-Saharan Africa",
    dat$region)
  unique(dat$region)
  return(dat)
}

full.dat.ssp2 <- add.reg.fun(full.dat.ssp2)
full.dat.ssp2.na <- add.reg.fun(full.dat.ssp2.na)

relreg <- function(dat) {
  dat$region <- factor(dat$region,
    levels = c("Middle East & North Africa",
      "Sub-Saharan Africa",
      "Asia", "Oceania",
      "Northern America",
      "Latin America and the Caribbean",
      "Europe, EU", "Europe, non-EU"))
}

full.dat.ssp2.na$region <- relreg(full.dat.ssp2.na)
```

Then we plot the regional breakdowns which show when broken down by region the model indicates both low error and bias.

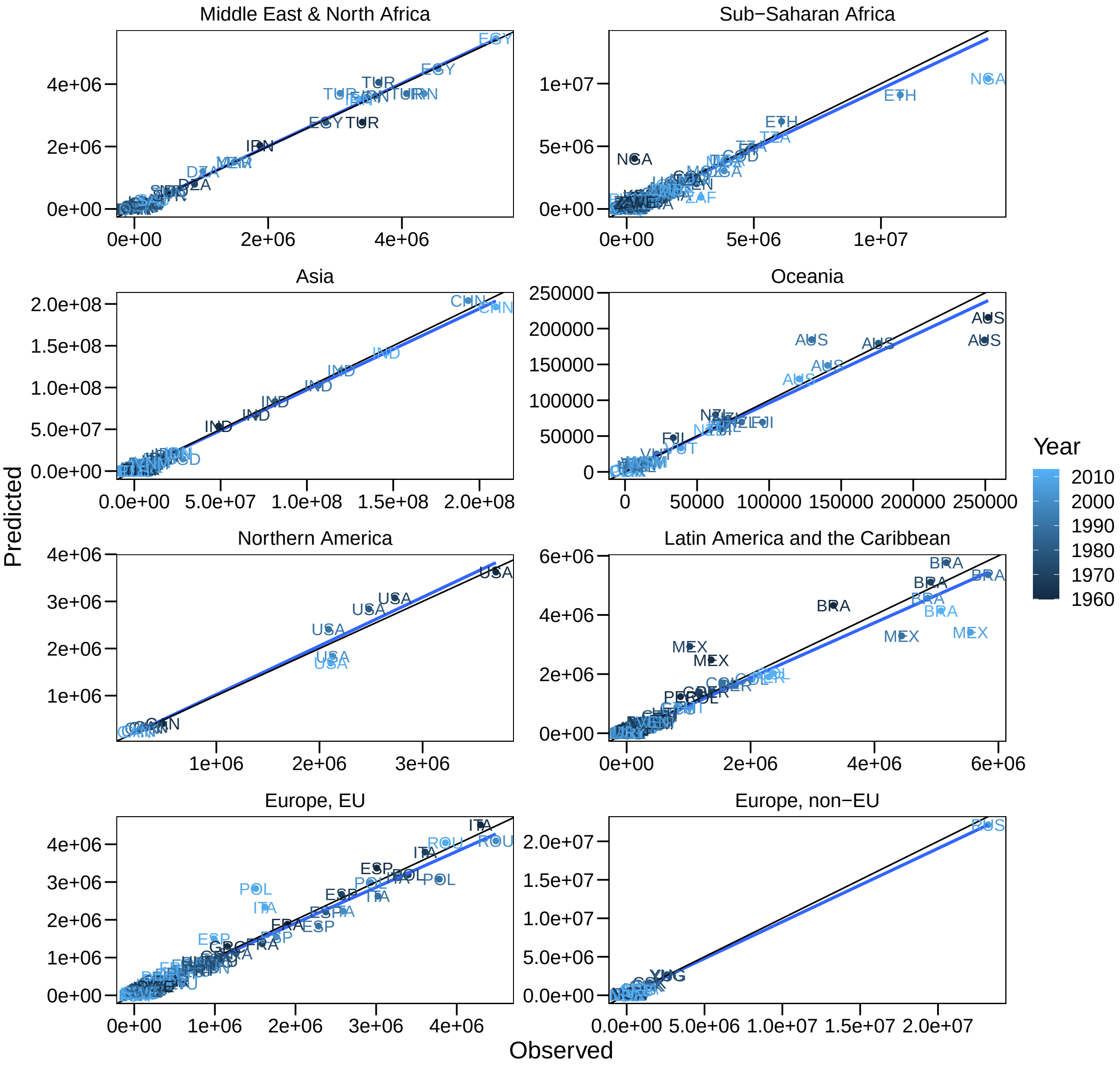
```r
pb <- ggplot(full.dat.ssp2.na,
  aes(log(count), (fit))) + geom_point() +
  geom_smooth(method = "lm",
    se = F) + geom_abline(slope = 1,
  intercept = 0) + xlab("Observed") +
  ylab("Predicted") + facet_wrap(~region,
  scales = "free", ncol = 2) +
  theme_base() + theme(plot.background = element_blank())
pb
```



```
ggsave(pb, filename = "Figs/FigS4.png",
  dpi = 300, width = 10.5, height = 10)
```

Finally, we check the predictions on the back-transformed scale. Main bias issue seems to be in Africa, under-predicting for example, for Ethiopia and Nigeria. Also note the bias in Asia, for recent years under-predicting China and India by around 35M collectively. This error should show up in uncertainty analysis.

```
pobr <- ggplot(full.dat.ssp2.na,
  aes((count), exp(fit), label = ISO3_CODE,
    color = Year)) + geom_point() +
  geom_smooth(method = MASS::"rlm",
    se = F) + geom_abline(slope = 1,
  intercept = 0) + xlab("Observed") +
  ylab("Predicted") + facet_wrap(~region,
  scales = "free", ncol = 2) +
  geom_text() + theme_base() +
  theme(plot.background = element_blank())
pobr
```



```
ggsave(pobr, filename = "Figs/FigS5.png",
  dpi = 300, width = 10.5, height = 10)
```

4.4.5 Rough inference

We then extremely rough inference on the smooths using a linear approximation across the range of observed data.

```
d <- plot(q2, allTerms = TRUE)

areaeffect <- (d$plots[[1]]$data$fit$y[100] -
  d$plots[[1]]$data$fit$y[1])/(d$plots[[1]]$data$fit$x[100] -
  d$plots[[1]]$data$fit$x[1])

gdpeffect <- (d$plots[[2]]$data$fit$y[100] -
  d$plots[[2]]$data$fit$y[1])/(d$plots[[2]]$data$fit$x[100] -
  d$plots[[2]]$data$fit$x[1])

popeffect <- (d$plots[[3]]$data$fit$y[100] -
  d$plots[[3]]$data$fit$y[1])/(d$plots[[3]]$data$fit$x[100] -
  d$plots[[3]]$data$fit$x[1])

areaeffect
```

```
## [1] 0.1423686
```

```
gdpeffect
```

```
## [1] -0.1362107
```

```
popeffect
```

```
## [1] 0.8279425
```

5 Model deployment

The model above is deployed on the time series of 189 countries for which agricultural area, GDP and rural population size where available to generate estimates of the number of farms per country for each year, along with 95% credible intervals. For countries included in the model building, adjustments in the predictions were made based on country level intercepts, for other countries outside the training data, population level predictions (i.e. averages across countries) were used.

These country level predictions were then summed to the regional and global level, to generate average estimates of farm number by region by year along with their 95% credible intervals of those averages. the model was also deployed to predict future farm numbers, as was conducted for the historical data, albeit holding agricultural area constant at 2013 levels; again summed to compute regional and global averages.

Finally, back transformed historical and future projections were passed through a mean filter (a LOESS regression of farm counts ~ year, with a span parameter of 0.75) to smooth noise from acute structural breaks in predictions (e.g. from financial crises, soviet break-down). Farm numbers under alternative SSPs were estimated using replicate methodology; as were different country development groupings.

5.1 Pre-processing

First I append basic development groupings based on gdp per capita thresholds.

```
dev.lookup <- full.dat.ssp2 %>%
  filter(Year == 2013) %>%
  mutate(gdpg = case_when(gdpcap.US2005 <
    1000 ~ "Low", 1000 <= gdpcap.US2005 &
    gdpcap.US2005 < 4000 ~
    "Lower Middle", 4000 <=
    gdpcap.US2005 & gdpcap.US2005 <
    13000 ~ "Upper Middle",
    13000 <= gdpcap.US2005 ~
      "High")) %>%
  select(ISO3_CODE, gdpg) %>%
  distinct()
```

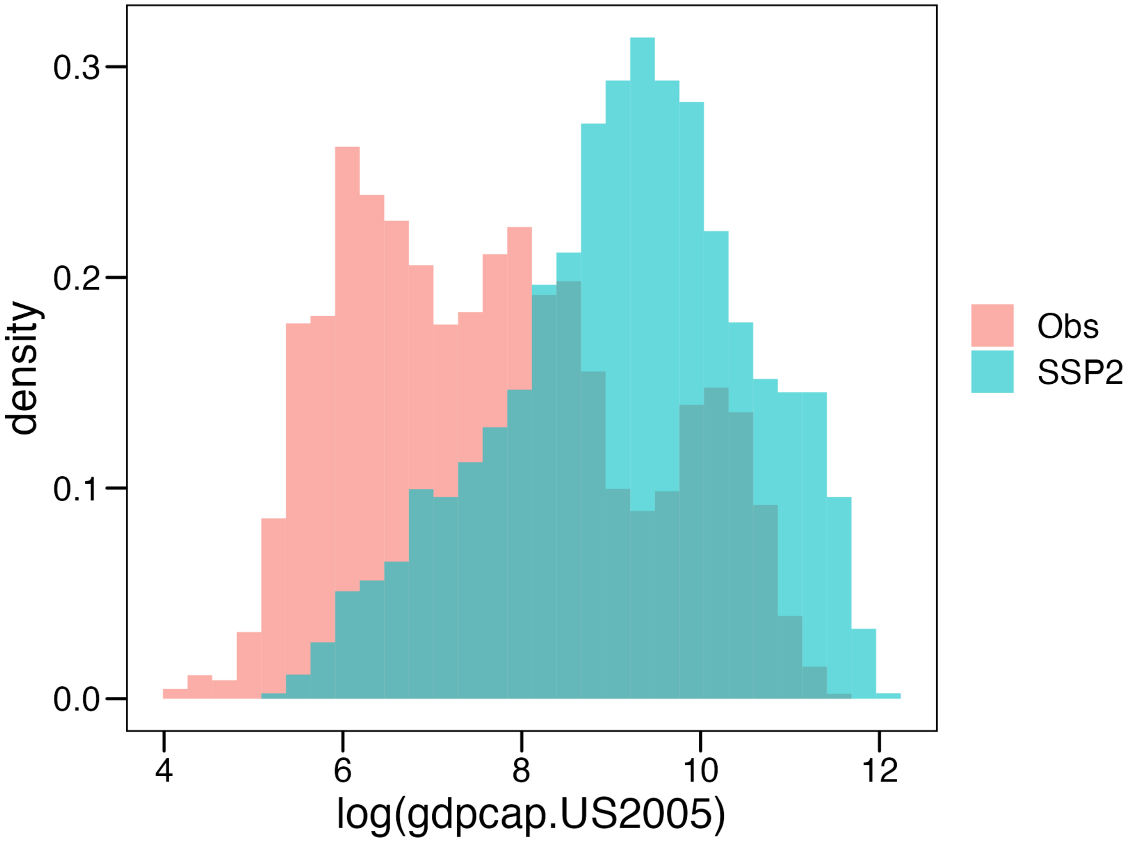
And remove small island states from the predictor dataframe

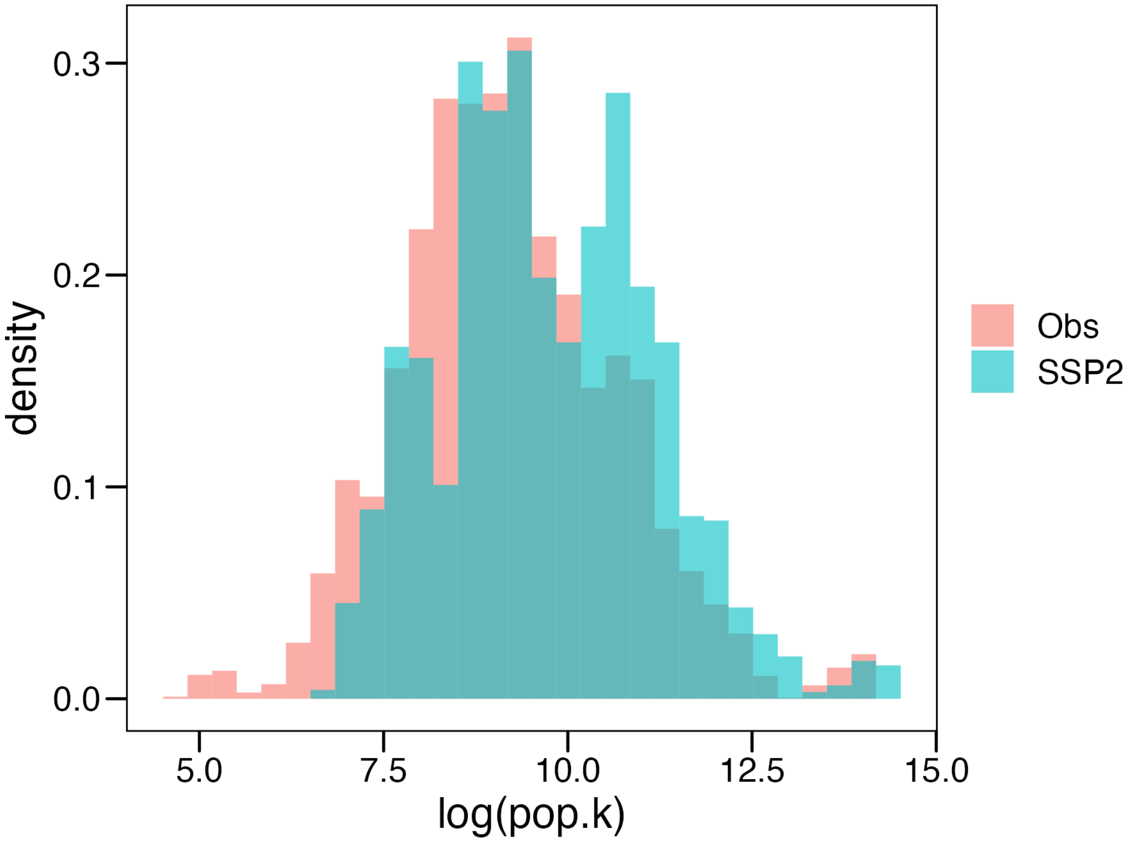
```
sisetc <- setdiff(unique(factor(full.dat.ssp1[which(full.dat.ssp1$Year ==
  2013), ]$ISO3_CODE)), unique(factor(full.dat.ssp1[which(full.dat.ssp1$Year ==
  2015), ]$ISO3_CODE)))

full.dat.ssp2.trim <- subset(full.dat.ssp2,
  !(ISO3_CODE %in% sisetc))
full.dat.ssp3.trim <- subset(full.dat.ssp3,
  !(ISO3_CODE %in% sisetc))
full.dat.ssp1.trim <- subset(full.dat.ssp1,
  !(ISO3_CODE %in% sisetc))

# checks
setdiff(unique(factor(full.dat.ssp2.trim[which(full.dat.ssp2.trim$Year ==
  2013), ]$ISO3_CODE)), unique(factor(full.dat.ssp2.trim[which(full.dat.ssp2.trim$Year ==
  2015), ]$ISO3_CODE)))
length(unique(full.dat.ssp2.trim$ISO3_CODE))
```

5.2 Extrapolation

For future predictions we check the histogram of the predicted vs realized gdp data and pop data to ensure we are not predicting far outside the model functional forms parameterized on the observed data.

```
full.dat.ssp2.trim$predictor.space <- ifelse(full.dat.ssp2.trim$Year <
  2013, "Obs", "SSP2")
densgdp <- ggplot(full.dat.ssp2.trim,
  aes(log(gdpcap.US2005), fill = predictor.space)) +
  geom_histogram(aes(y = ..density..),
    position = position_identity(),
    alpha = 0.6) + theme_base() +
  theme(legend.position = "right",
    legend.margin = margin(-0.8,
      0, 0, 0, unit = "cm"),
    legend.title = element_blank()) +
  theme(plot.background = element_blank())

denspop <- ggplot(full.dat.ssp2.trim,
  aes(log(pop.k), fill = predictor.space)) +
  geom_histogram(aes(y = ..density..),
    position = position_identity(),
    alpha = 0.6) + theme_base() +
  theme(legend.position = "right",
    legend.margin = margin(-0.8,
      0, 0, 0, unit = "cm"),
    legend.title = element_blank()) +
  theme(plot.background = element_blank())

g <- gridExtra::arrangeGrob(densgdp,
  denspop, nrow = 2)  #generates g
ggsave(file = "Figs/FigS6.png",
  g, dpi = 300, width = 6, height = 9)  #saves g
g <- image_read("Figs/FigS6.png")
g
```


0.3
0.2
0.1
0.0
density
4
6
8
10
12
log(gdpcap.US2005)
Obs
SSP2


0.3
0.2
0.1
0.0
density
5.0
7.5
10.0
12.5
15.0
log(pop.k)
Obs
SSP2

5.3 Predictions

Then we write a function to perform the historical reconstruction as well as the future predictions of farm number, with aggregation of numbers and credible intervals on the global, regional, and development grouping levels. We smooth over the predictions to further join the SSP projections with the observed record (this mismatch doesn't show on the global level, but regionally it seems to be upward or downward biased, the reason is the countries not covered by the SSP that are in the observed record, note this bias is very small in general, based on uncertainty of the overall predictions this is of little importance, at least regionally or globally). This smoothing also deals with structural short term changes in gdp, e.g. due to financial breakdowns, and also somewhat to the breakdown of the Soviet Union, which led to an increase in farm holdings in Europe.

```r
getpred <- function(dat, mod, span) {
  dat <- add.reg.fun(dat)
  dat <- left_join(dat, dev.lookup,
    by = "ISO3_CODE")
  dat$gdpg <- as.factor(dat$gdpg)
  dat$log.gdp <- log(dat$gdpcap.US2005)
  dat$log.area <- log(dat$Agarea.kha)
  dat$log.rural.pop <- log(dat$pop.rural.k)

  newdf <- as.data.frame(subset(dat,
    Year > 1969))
  pn <- predict(mod, newdf, se = T)
  newdf <- cbind(pn, newdf)

  # dev grouping
  pred.time.dev <- newdf %>%
    filter(gdpg != "NA", Year >
      2013) %>%
    group_by(Year, gdpg) %>%
    summarize(Median = sum(exp(fit),
      na.rm = T), upper = sum(exp(fit +
      se.fit * 2), na.rm = T),
      lower = sum(exp(fit -
        se.fit * 2), na.rm = T)) %>%
    ungroup() %>%
    group_by(gdpg) %>%
    mutate(Median.smooth = fitted(loess(Median ~
      Year, span = span)),
      upper.smooth = fitted(loess(upper ~
        Year, span = span)),
      lower.smooth = fitted(loess(lower ~
        Year, span = span)))

  # reg grouping
  pred.time.reg <- newdf %>%
    group_by(Year, region) %>%
    summarize(Median = sum(exp(fit),
      na.rm = T), upper = sum(exp(fit +
      se.fit * 2), na.rm = T),
      lower = sum(exp(fit -
        se.fit * 2), na.rm = T)) %>%
    ungroup() %>%
    group_by(region) %>%
    mutate(Median.smooth = fitted(loess(Median ~
      Year, span = span)),
      upper.smooth = fitted(loess(upper ~
        Year, span = span)),
      lower.smooth = fitted(loess(lower ~
        Year, span = span)))

  # global
  pred.time <- newdf %>%
    group_by(Year) %>%
    summarize(Median = sum(exp(fit),
      na.rm = T), upper = sum(exp(fit +
      se.fit * 2), na.rm = T),
      lower = sum(exp(fit -
        se.fit * 2), na.rm = T)) %>%
    ungroup() %>%
    mutate(Median.smooth = fitted(loess(Median ~
      Year, span = span)),
      upper.smooth = fitted(loess(upper ~
        Year, span = span)),
      lower.smooth = fitted(loess(lower ~
        Year, span = span)))

  return(list(pred.time.reg,
    pred.time, pred.time.dev))

}
```

Then we run that function over each SSP of interest.

```
ssp2 <- getpred(full.dat.ssp2.trim,
  q2, 0.75)
ssp3 <- getpred(full.dat.ssp3.trim,
  q2, 0.75)
ssp1 <- getpred(full.dat.ssp1.trim,
  q2, 0.75)

ssp2[[1]]$region <- relreg(ssp2[[1]])
ssp1[[1]]$region <- relreg(ssp1[[1]])
ssp3[[1]]$region <- relreg(ssp3[[1]])
```

5.4 Regional summary

Here is a table of predicted farm numbers and size by region between the 2020, 2050 and 2100 for reference. CI's not shown.

```
statreg <- ssp2[[1]] %>%
  group_by(region, Year) %>%
  select(1:2, 6) %>%
  filter(Year %in% c(2013, 2020,
    2050, 2100)) %>%
  arrange(region) %>%
  mutate_at(vars(Median.smooth),
    funs(round((./1e+06), digits = 1)))

r <- full.dat.ssp2.trim
r$region <- relreg(full.dat.ssp2.trim)
r <- r %>%
  filter(Year == 2100) %>%
  group_by(region) %>%
  summarize(area = sum(Agarea.kha)) %>%
  left_join(statreg) %>%
  mutate(`mean farmsize (Ha)` = round((area *
    1000)/(Median.smooth *
    1e+06)), digits = 1) %>%
  select(1, 3:5) %>%
  rename(., `farm number (M)` = Median.smooth,
    year = "Year")

kableExtra::kbl(r)
```

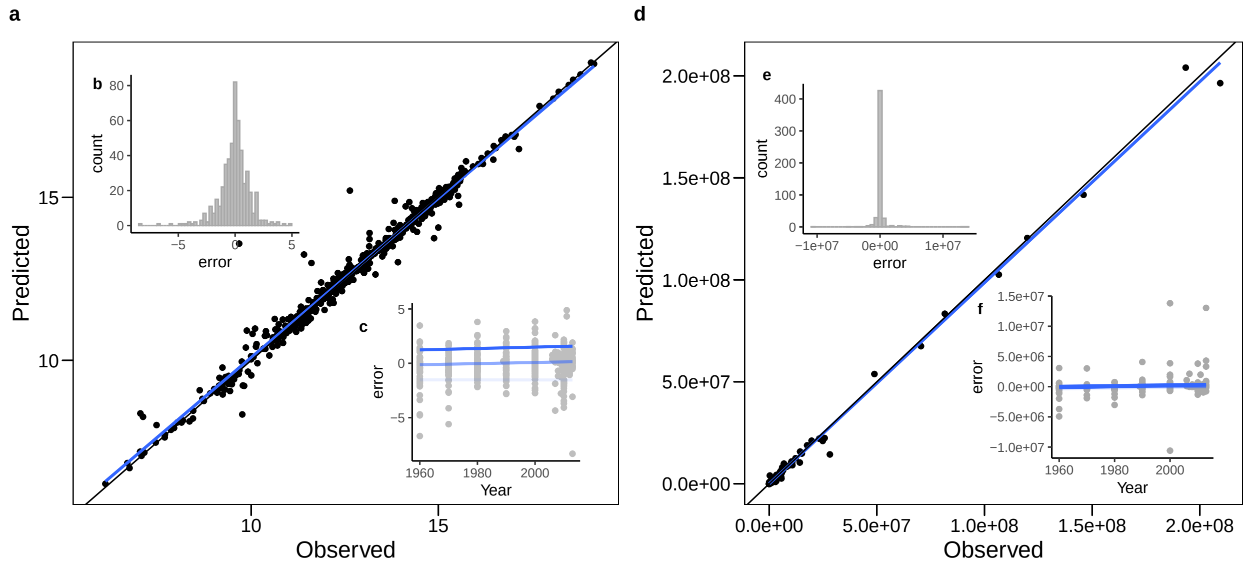
region	year	farm number (M)	mean farmsize (Ha)
Middle East & North Africa	2013	25.6	20
Middle East & North Africa	2020	26.7	19
Middle East & North Africa	2050	27.0	19
Middle East & North Africa	2100	14.4	36
Sub-Saharan Africa	2013	83.5	11
Sub-Saharan Africa	2020	93.8	10
Sub-Saharan Africa	2050	129.0	7
Sub-Saharan Africa	2100	115.3	8
Asia	2013	437.3	3
Asia	2020	438.0	3
Asia	2050	367.9	4
Asia	2100	131.4	10
Oceania	2013	0.7	584
Oceania	2020	0.8	511
Oceania	2050	0.9	454
Oceania	2100	0.5	818
Northern America	2013	1.8	261
Northern America	2020	1.6	294
Northern America	2050	0.9	523
Northern America	2100	0.3	1569
Latin America and the Caribbean	2013	18.0	42
Latin America and the Caribbean	2020	17.5	43
Latin America and the Caribbean	2050	12.9	58
Latin America and the Caribbean	2100	4.3	175
Europe, EU	2013	14.9	12
Europe, EU	2020	13.5	14
Europe, EU	2050	7.2	26
Europe, EU	2100	2.1	89
Europe, non-EU	2013	26.3	11
Europe, non-EU	2020	24.2	12
Europe, non-EU	2050	14.4	20
Europe, non-EU	2100	4.2	67

6 Main Text Figures

Below I recreate the main assets used in the main text of the manuscript. This include the main figures (also stored in the 'Figs' directory), as well as the data underlying them for the middle of the road scenario for reference (stored in the 'Data/OUTPUT/' directory).

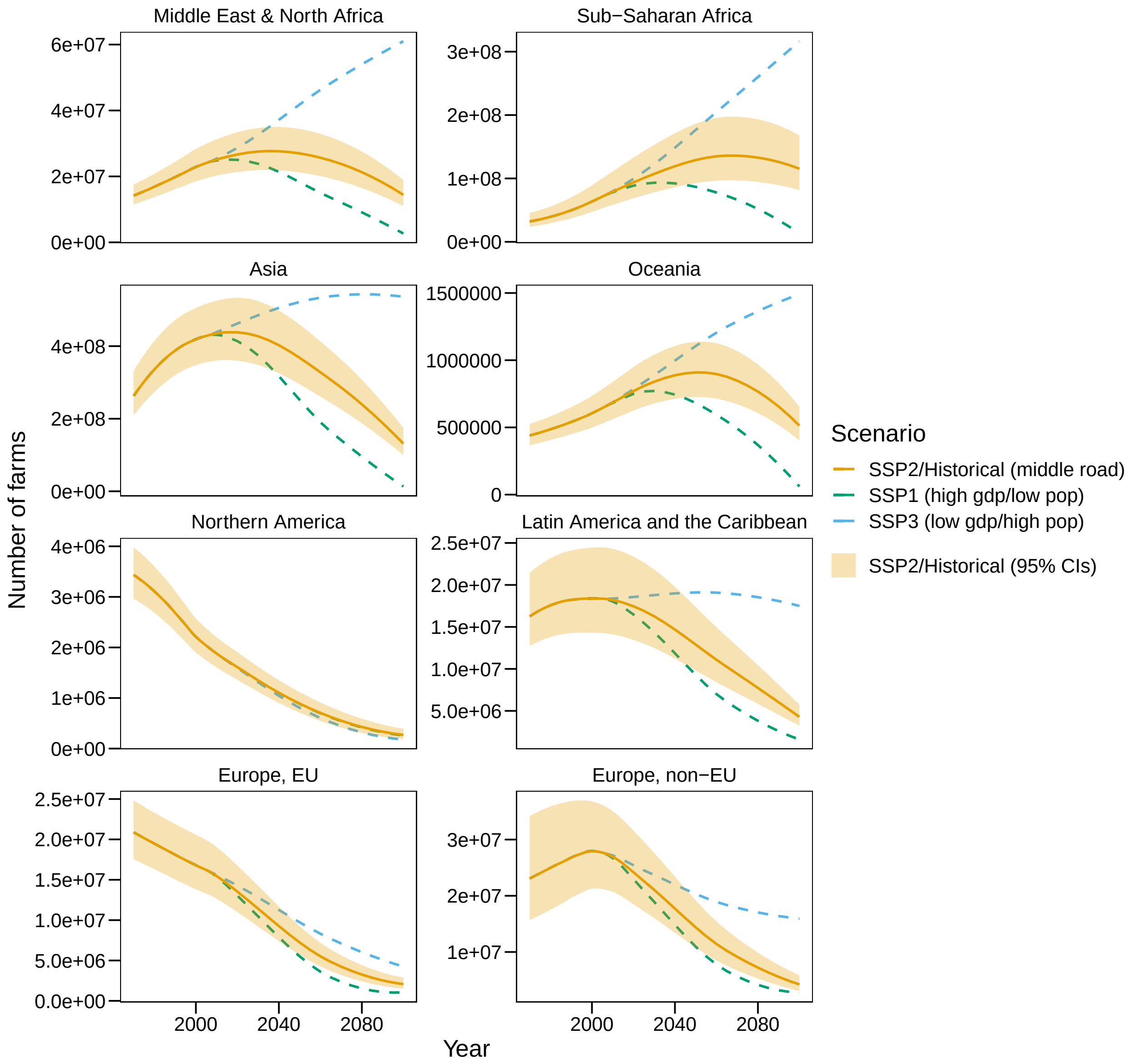
6.1 Figure 1

```
p
```



6.2 Figure 2

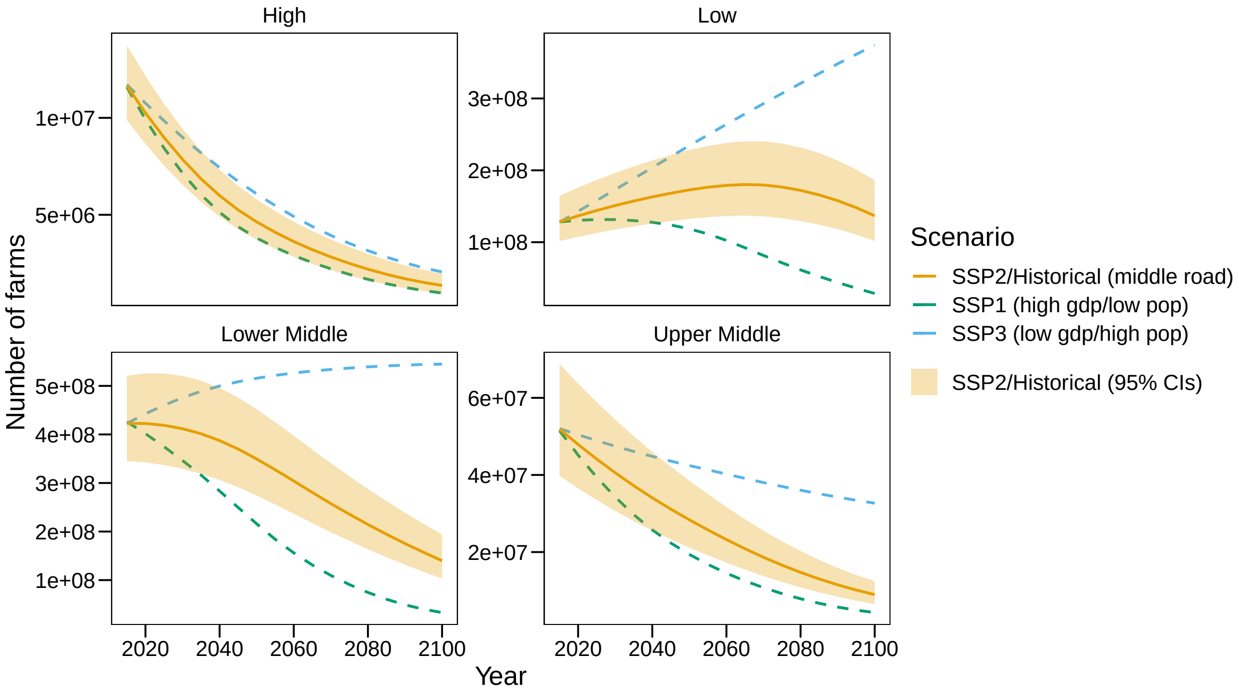
```r
cols <- palette.colors(palette = "Okabe-Ito")
palette(cols)
regional <- ggplot(ssp2[[1]], aes(Year,
  Median.smooth)) + geom_line(aes(ssp1[[1]]$Year,
  ssp1[[1]]$Median.smooth, col = "SSP1 (high gdp/low pop)"),
  lwd = 0.8, lty = 2, alpha = 1) +
  geom_line(aes(ssp3[[1]]$Year,
    ssp3[[1]]$Median.smooth,
    col = "SSP3 (low gdp/high pop)"),
    lwd = 0.8, lty = 2, alpha = 1) +
  geom_ribbon(aes(ymin = lower.smooth,
    ymax = upper.smooth, fill = "SSP2/Historical (95% CIs)"),
    alpha = 0.3) + geom_line(aes(col = "SSP2/Historical (middle road)"),
  lwd = 0.8) + ylab("Number of farms ") +
  theme_base() + scale_color_discrete(breaks = c("SSP2/Historical (middle road)",
  "SSP1 (high gdp/low pop)",
  "SSP3 (low gdp/high pop)"),
  type = palette()[c(4, 2, 3)],
  name = "Scenario") + scale_fill_discrete(type = palette()[2],
  name = "") + guides(col = guide_legend(order = 1),
  fill = guide_legend(order = 2)) +
  facet_wrap(~region, scales = "free_y",
    ncol = 2) + theme(legend.position = "right",
  legend.margin = margin(-0.8,
    0, 0, 0, unit = "cm")) +
  theme(plot.background = element_blank())
regional
```



```
ggsave(regional, filename = "Figs/Fig2.png",
  dpi = 300, width = 10.5, height = 10)

ssp2[[1]] %>%
  select(Median.smooth, lower.smooth,
    upper.smooth) %>%
  write.csv(., "Data/OUTPUT/Fig2SSP2.csv")
```

6.3 Figure 3

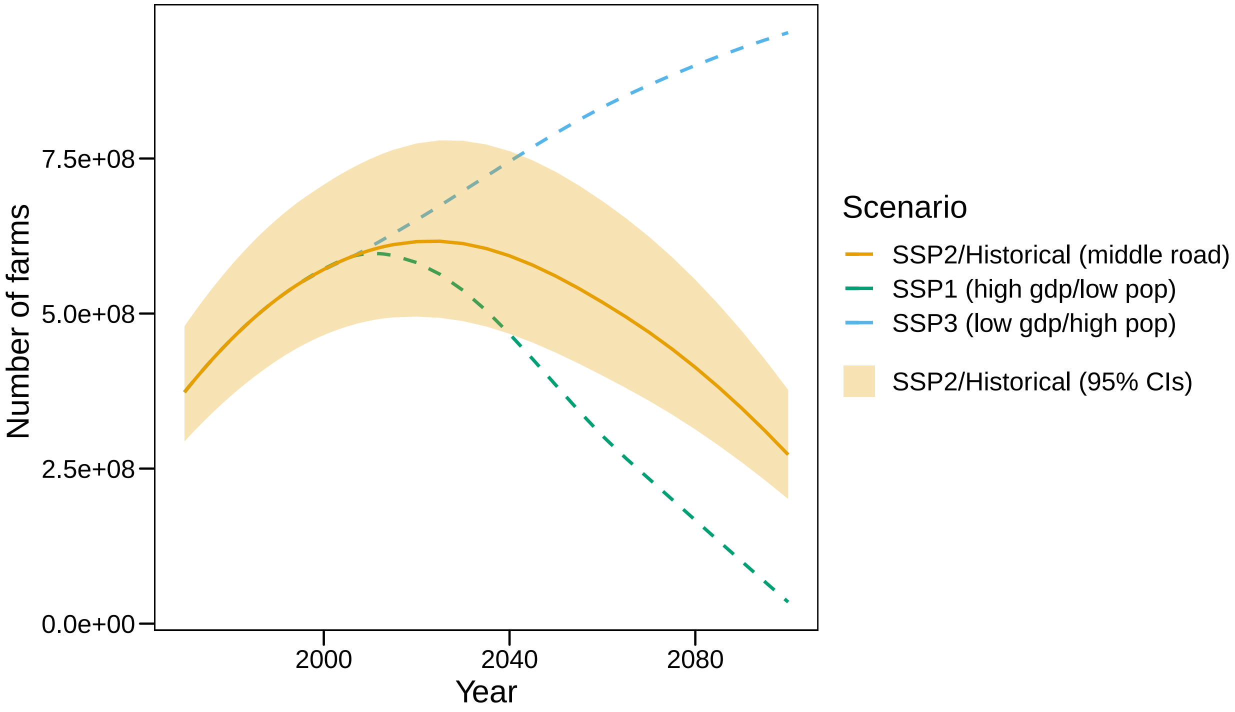
```
devgroup <- ggplot(ssp2[[3]], aes(Year,
  Median.smooth)) + geom_line(aes(ssp1[[3]]$Year,
  ssp1[[3]]$Median.smooth, col = "SSP1 (high gdp/low pop)"),
  lwd = 0.8, lty = 2, alpha = 1) +
  geom_line(aes(ssp3[[3]]$Year,
    ssp3[[3]]$Median.smooth,
    col = "SSP3 (low gdp/high pop)"),
    lwd = 0.8, lty = 2, alpha = 1) +
  geom_ribbon(aes(ymin = lower.smooth,
    ymax = upper.smooth, fill = "SSP2/Historical (95% CIs)"),
    alpha = 0.3) + geom_line(aes(col = "SSP2/Historical (middle road)"),
  lwd = 0.8) + ylab("Number of farms ") +
  theme_base() + scale_color_discrete(breaks = c("SSP2/Historical (middle road)",
  "SSP1 (high gdp/low pop)",
  "SSP3 (low gdp/high pop)"),
  type = palette()[c(4, 2, 3)],
  name = "Scenario") + scale_fill_discrete(type = palette()[2],
  name = "") + guides(col = guide_legend(order = 1),
  fill = guide_legend(order = 2)) +
  facet_wrap(~gdpg, scales = "free_y",
    ncol = 2) + theme(legend.position = "right",
  legend.margin = margin(-0.8,
    0, 0, 0, unit = "cm")) +
  theme(plot.background = element_blank())
devgroup
```



```
ggsave(devgroup, filename = "Figs/Fig3.png",
  dpi = 300, width = 10.5, height = 6)

ssp2[[3]] %>%
  select(Median.smooth, lower.smooth,
    upper.smooth) %>%
  write.csv(., "Data/OUTPUT/Fig3SSP2.csv")
```

6.4 Figure 4

```
palette(cols)
global <- ggplot(ssp2[[2]], aes(Year,
  Median.smooth)) + geom_line(aes(ssp1[[2]]$Year,
  ssp1[[2]]$Median.smooth, col = "SSP1 (high gdp/low pop)"),
  lwd = 0.8, lty = 2, alpha = 1) +
  geom_line(aes(ssp3[[2]]$Year,
    ssp3[[2]]$Median.smooth,
    col = "SSP3 (low gdp/high pop)"),
    lwd = 0.8, lty = 2, alpha = 1) +
  geom_ribbon(aes(ymin = lower.smooth,
    ymax = upper.smooth, fill = "SSP2/Historical (95% CIs)"),
    alpha = 0.3) + geom_line(aes(col = "SSP2/Historical (middle road)"),
  lwd = 0.8) + ylab("Number of farms ") +
  theme_base() + scale_color_discrete(breaks = c("SSP2/Historical (middle road)",
  "SSP1 (high gdp/low pop)",
  "SSP3 (low gdp/high pop)"),
  type = palette()[c(4, 2, 3)],
  name = "Scenario") + scale_fill_discrete(type = palette()[2],
  name = "") + guides(col = guide_legend(order = 1),
  fill = guide_legend(order = 2)) +
  theme(legend.position = "right",
    legend.margin = margin(-0.8,
      0, 0, 0, unit = "cm")) +
  theme(plot.background = element_blank())
global
```



```
ggsave(global, filename = "Figs/Fig4.png",
  dpi = 300, width = 8.75, height = 5.11)

ssp2[[2]] %>%
  select(Median.smooth, lower.smooth,
    upper.smooth) %>%
  write.csv(., "Data/OUTPUT/Fig4SSP2.csv")
```

7 Session info

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] checkpoint_1.0.2  knitr_1.41        gridExtra_2.3     kableExtra_1.3.4
##  [5] zoo_1.8-12        countrycode_1.4.0 readxl_1.4.1      formatR_1.14
##  [9] ggokabeito_0.1.0  magick_2.7.3      ggthemes_4.2.4    ggpubr_0.5.0
## [13] mgcViz_0.1.9      qgam_1.3.4        mgcv_1.8-40       nlme_3.1-157
## [17] FAOSTAT_2.2.3     ggplot2_3.4.0     tidyr_1.2.1       dplyr_1.0.10
##
## loaded via a namespace (and not attached):
##  [1] minqa_1.2.5        colorspace_2.0-3   ggsignif_0.6.4     ellipsis_0.3.2
##  [5] class_7.3-20       rstudioapi_0.14    proxy_0.4-27       farver_2.1.1
##  [9] MatrixModels_0.5-1 fansi_1.0.3        xml2_1.3.3         codetools_0.2-18
## [13] splines_4.2.1      doParallel_1.0.17  nloptr_2.0.3       broom_1.0.1
## [17] shiny_1.7.3        compiler_4.2.1     httr_1.4.4         backports_1.4.1
## [21] assertthat_0.2.1   Matrix_1.5-3       fastmap_1.1.0      cli_3.4.1
## [25] later_1.3.0        htmltools_0.5.3    quantreg_5.94      tools_4.2.1
## [29] gtable_0.3.1       glue_1.6.2         Rcpp_1.0.9         carData_3.0-5
## [33] cellranger_1.1.0   vctrs_0.5.1        RJSONIO_1.3-1.6    svglite_2.1.0
## [37] iterators_1.0.14   xfun_0.38          stringr_1.4.1      lme4_1.1-31
## [41] rvest_1.0.3        mime_0.12          miniUI_0.1.1.1     lifecycle_1.0.3
## [45] rstatix_0.7.1      MASS_7.3-57        scales_1.2.1       ragg_1.2.4
## [49] promises_1.2.0.1   parallel_4.2.1     SparseM_1.81       RColorBrewer_1.1-3
## [53] gamm4_0.2-6        yaml_2.3.6         reshape_0.8.9      stringi_1.7.8
## [57] highr_0.9          foreach_1.5.2      e1071_1.7-12       boot_1.3-28
## [61] rlang_1.0.6        pkgconfig_2.0.3    systemfonts_1.0.4  matrixStats_0.63.0
## [65] evaluate_0.18      lattice_0.20-45    purrr_0.3.5        labeling_0.4.2
## [69] cowplot_1.1.1      tidyselect_1.2.0   GGally_2.1.2       plyr_1.8.8
## [73] magrittr_2.0.3     R6_2.5.1           generics_0.1.3     DBI_1.1.3
## [77] pillar_1.8.1       withr_2.5.0        survival_3.3-1     abind_1.4-5
## [81] tibble_3.1.8       car_3.1-1          KernSmooth_2.23-20 utf8_1.2.2
## [85] rmarkdown_2.18     viridis_0.6.2      grid_4.2.1         data.table_1.14.6
## [89] digest_0.6.30      classInt_0.4-8     webshot_0.5.4      xtable_1.8-4
## [93] httpuv_1.6.6       textshaping_0.3.6  munsell_0.5.0      viridisLite_0.4.1
```

References

Arel-Bundock, Vincent. 2022. *Countrycode: Convert Country Names and Country Codes.* https://vincentarelbundock.github.io/countrycode/.

Arnold, Jeffrey B. 2021. *Ggthemes: Extra Themes, Scales and Geoms for Ggplot2.* https://github.com/jrnold/ggthemes.

Auguie, Baptiste. 2017. *gridExtra: Miscellaneous Functions for "Grid" Graphics.* https://CRAN.R-project.org/package=gridExtra.

Barrett, Malcolm. 2021. *Ggokabeito: Okabe-Ito Scales for Ggplot2 and Ggraph.* https://CRAN.R-project.org/package=ggokabeito.

Fasiolo, Matteo. 2021. *Qgam: Smooth Additive Quantile Regression Models.* https://CRAN.R-project.org/package=qgam.

Fasiolo, Matteo, and Raphael Nedellec. 2021. *mgcViz: Visualisations for Generalized Additive Models.* https://github.com/mfasiolo/mgcViz.

Kao, Michael C. J., Markus Gesmann, and Filippo Gheri. 2022. *FAOSTAT: Download Data from the FAOSTAT Database.* https://gitlab.com/paulrougieux/faostatpackage.

Kassambara, Alboukadel. 2022. *Ggpubr: Ggplot2 Based Publication Ready Plots.* https://rpkgs.datanovia.com/ggpubr/.

Lowder, Sarah K., Marco V. Sánchez, and Raffaele Bertini. 2021. "Which Farms Feed the World and Has Farmland Become More Concentrated?" *World Development.* https://doi.org/10.1016/j.worlddev.2021.105455.

Mehrabi, Zia. 2023a. "Likely Decline in the Number of Farms Globally by the Middle of the Century." *Nature Sustainability.* https://doi.org/10.1038/s41893-023-01110-y.

———. 2023b. "Likely Decline in the Number of Farms Globally by the Middle of the Century." *SocArXiv.* https://doi.org/10.31235/osf.io/m6p5u.

Ooi, Hong, Andrie de Vries, and Microsoft. 2022. *Checkpoint: Install Packages from Snapshots on the Checkpoint Server for Reproducibility.* https://github.com/RevolutionAnalytics/checkpoint.

Ooms, Jeroen. 2021. *Magick: Advanced Graphics and Image-Processing in r.* https://CRAN.R-project.org/package=magick.

Wickham, Hadley, and Jennifer Bryan. 2022. *Readxl: Read Excel Files.* https://CRAN.R-project.org/package=readxl.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2022. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. *Dplyr: A Grammar of Data Manipulation.* https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Maximilian Girlich. 2022. *Tidyr: Tidy Messy Data.* https://CRAN.R-project.org/package=tidyr.

Xie, Yihui. 2022. *Knitr: A General-Purpose Package for Dynamic Report Generation in r.* https://yihui.org/knitr/.

———. 2023. *formatR: Format r Code Automatically.* https://github.com/yihui/formatR.

Zeileis, Achim, Gabor Grothendieck, and Jeffrey A. Ryan. 2023. *Zoo: S3 Infrastructure for Regular and Irregular Time Series (z's Ordered Observations).* https://zoo.R-Forge.R-project.org/.

Zhu, Hao. 2021. *kableExtra: Construct Complex Table with Kable and Pipe Syntax.* https://CRAN.R-project.org/package=kableExtra.