The driving factors behind food prices in developing countries

Benjamin Jaidi
Stefan Ramakrishnan
Raiber Alkurdi

Ladislaus von Bortkiewicz Chair of Statistics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de

Outline

General

According to OECD (rising food prices: causes and consequences 2008):
Price of major crops are affected by

- Production below trend
- Growth of demand
- Low stocks levels
- Investments in derivative agricultural markets
- Humanitarian aid could help strengthen poor consumers -> influence on price?
- Subsidies in agriculture to increase supply
- Production of biofuels

General

- Can we confirm or reject this? Let the data talk for themselves!
- Analysis of historic data from selected countries
- Example: 4 most important crops in India based on produced amount per year
    - Wheat, Potatoes, Sugarcane, Rice

Explanatory Variables

Supply related factors:

- Climate data:
  - Amount of rain per year, temperature
- Production:
  - Oil price
  - Produced amount per crop per year
- Macroeconomic:
  - GDP in agriculture
  - Inflation consumer prices
  - Import of goods from HS2017 06-15

Demand related factors

- Demographic:
  - GNI per capita
  - Per capita calorie intake
  - Population size
- Macroeconomic:
  - Exports of goods from HS2017 06-15

Overview

After cleansing, merging and feature construction our dataset looks like this

```
> head(india, n=5)
  year cm_name adm0_name um_id um_name avg_price_prod_year     pr_q1    pr_q2    pr_q3    pr_q4   tas_q1
1 2001    Rice     India     5      KG           10.264951  7.778937 90.32667 170.5643 35.64556 20.62653
2 2001   Sugar     India     5      KG           16.327288  7.778937 90.32667 170.5643 35.64556 20.62653
3 2001   Wheat     India     5      KG            8.399091  7.778937 90.32667 170.5643 35.64556 20.62653
4 2002    Rice     India     5      KG           10.491068 12.721033 74.45227 161.2643 30.19737 20.66723
5 2002   Sugar     India     5      KG           15.875177 12.721033 74.45227 161.2643 30.19737 20.66723
    tas_q2   tas_q3   tas_q4 prod_amount_y daily_caloric_supply  exp_sug  exp_veg   exp_cer   imp_sug
1 28.73153 26.92260 22.03897         93340                 2333 360063.1 851221.6 360063109 26256.35
2 28.73153 26.92260 22.03897        297208                 2333 360063.1 851221.6 360063109 26256.35
3 28.73153 26.92260 22.03897         72766                 2333 360063.1 851221.6 360063109 26256.35
4 29.71300 27.01523 22.01637         71820                 2285 321010.0 900122.0  32101005 26033.65
5 29.71300 27.01523 22.01637        287383                 2285 321010.0 900122.0  32101005 26033.65
  imp_veg  imp_cer     agri_gdp gni_pc cp_inflation avg_p_barrel population
1 1140825 25910.95 224032774207 778.43     3.684807        23.12 1071477855
2 1140825 25910.95 224032774207 778.43     3.684807        23.12 1071477855
3 1140825 25910.95 224032774207 778.43     3.684807        23.12 1071477855
4 1341905 28076.32 209237124425 796.12     4.392200        24.36 1089807112
5 1341905 28076.32 209237124425 796.12     4.392200        24.36 1089807112
>
```

Examples

- Product prices where given per market per month:
  - We calculated average per country per year
- Weather and rain data where given per month:
  - We created one feature per quarter (mean over three months)
- Calories per years had NAs:
  - We imputed values based on the average of the last 5 years

Overview

Loading the data and selecting the explanatory variables and target

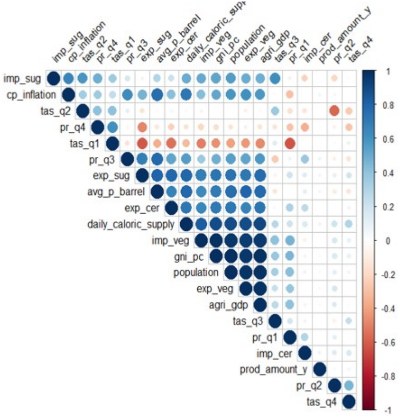
```
# initial variable selection and normalization
colselection = c("avg_price_prod_year","pr_q1","pr_q2","pr_q3","pr_q4","tas_q1","tas_q2","tas_q3","tas_q4","prod_amount_y","daily_caloric_supply","exp_sug","exp_veg","exp_cer","imp_sug","imp_veg","imp_cer","agri_gdp","gni_pc","cp_inflation","avg_p_barrel","population")
target = c("avg_price_prod_year")
normalized = as.data.frame(scale(india[colselection]))
feats = normalized[, !(colnames(normalized) %in% target)]
```

Naive approach

```
# Variable selection and modeling
# Model with all explanatory variables
# print corrplot for all explanatory variables
foo_insign = cor( feats, method = "pearson", use = "complete.obs")
corrplot(foo_insign, type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```

Naive approach

Many correlated explanatory variables!



Naive approach

```
# build model with all explanatory variables
expvarsall = paste(colnames(feats), collapse = '+')
formulaall = paste(target,"~", expvarsall, collapse
  = " ")
mod_varall = summary(lm(formulaall ,data =
  normalized))
# Regression coefficients have NAs, high R^2 but few
    signifikant predictors
# => indicator for multicolinearity
# we should remove some explanatory variables!
```

Naive approach

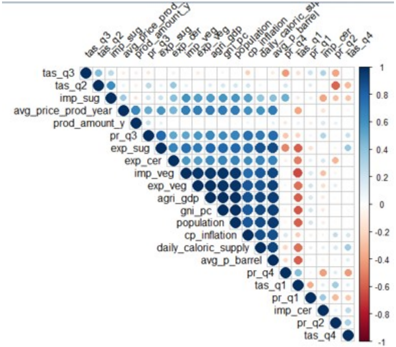
As expected we don't get a good result: Some correlation coefficients couldn't be calculated Only one significant variable low p-value, high R^2. This could be due to multicolinearity

```
Call:
lm(formula = formulaall, data = normalized)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.66128 -0.22110  0.02984  0.20824  0.67537 

Coefficients: (6 not defined because of singularities)
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -2.680e-15  4.994e-02   0.000    1.000    
pr_q1             -5.155e-01  3.962e-01  -1.301    0.202    
pr_q2              1.629e-02  1.711e-01   0.095    0.925    
pr_q3              4.260e-01  4.440e-01   0.959    0.344    
pr_q4             -3.998e-02  4.079e-01  -0.098    0.923    
tas_q1            -7.201e-01  6.347e-01  -1.135    0.265    
tas_q2             3.050e-01  2.690e-01   1.134    0.265    
tas_q3            -3.399e-02  3.671e-01  -0.093    0.927    
tas_q4             3.450e-01  2.901e-01   1.189    0.243    
prod_amount_y      6.330e-01  5.094e-02  12.425 5.42e-14 ***
daily_caloric_supply -1.005e-01  2.872e-01  -0.350    0.729    
exp_sug           -9.623e-01  1.153e+00  -0.834    0.410    
exp_veg            5.747e-01  4.473e-01   1.285    0.208    
exp_cer            9.880e-02  1.623e-01   0.609    0.547    
imp_sug            1.859e-02  3.788e-01   0.049    0.961    
imp_veg            5.094e-01  5.407e-01   0.942    0.353    
imp_cer                   NA         NA      NA       NA    
agri_gdp                  NA         NA      NA       NA    
gni_pc                    NA         NA      NA       NA    
cp_inflation              NA         NA      NA       NA    
avg_p_barrel              NA         NA      NA       NA    
population                NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3496 on 33 degrees of freedom
Multiple R-squared:  0.916,	Adjusted R-squared:  0.8778 
F-statistic: 23.99 on 15 and 33 DF,  p-value: 1.684e-13
```

Based on correlation significance

```
# Model with significant explanatory variables
# check pvalue for correlation target <=> explanatory variables (alpha = 0,05)
boo = rcorr(as.matrix(normalized))
cors <- as.data.frame(boo$r)
pvals = as.data.frame(boo$P)
pvalsr = pvals[pvals$avg_price_prod_year < 5*10^-2,]
vars = rownames(pvalsr)
vars = vars[vars != "NA"]
foo = cor(normalized[colnames(feats) %in% vars, ], method = "pearson", use = "complete.obs")
corrplot(foo, type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```

Based on correlation significance



Based on correlation significance

No more NAs in coefficients, still only one significant variable in lm model, still multicorrelated variables.

```r
# build model with significant explanatory variables only
expvarssig = paste(vars, collapse = "+")
formulasig = paste(target,"~",expvarssig, collapse ="+")
mod_varsig = summary(lm(formulasig,data = normalized))
```

```
Call:
lm(formula = formulasig, data = normalized)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.78560 -0.23293  0.03354  0.20513  0.63089 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          9.530e-16  5.046e-02   0.000    1.000    
pr_q3               -1.828e-01  1.948e-01  -0.938    0.355    
prod_amount_y        6.303e-01  5.143e-02  12.255 3.22e-14 ***
daily_caloric_supply -3.844e-01  2.546e-01  -1.510    0.140    
exp_sug              1.372e-01  4.740e-01   0.289    0.774    
exp_veg              1.775e-02  1.091e+00   0.016    0.987    
exp_cer             -3.699e-02  1.092e-01  -0.339    0.737    
imp_sug              1.818e-01  3.172e-01   0.573    0.570    
imp_veg             -1.270e-01  9.187e-01  -0.138    0.891    
agri_gdp             4.961e-01  8.211e-01   0.604    0.550    
gni_pc               1.240e+00  2.252e+00   0.551    0.585    
cp_inflation         1.350e-01  3.477e-01   0.388    0.700    
avg_p_barrel         2.381e-01  4.280e-01   0.556    0.582    
population          -9.314e-01  1.462e+00  -0.637    0.528    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3532 on 35 degrees of freedom
Multiple R-squared:  0.909,	Adjusted R-squared:  0.8753 
F-statistic: 26.91 on 13 and 35 DF,  p-value: 2.203e-14
```

Small model

Prices are usually determined by supply and demand. What if we only use two regressants:

- Supply(produced amount)
- demand (population size)

```
# model with produced amount and population only
mod_varsmall = summary(lm(avg_price_prod_year ~ +
  prod_amount_y + population ,data = normalized))
```

Small model

```
Call:
lm(formula = avg_price_prod_year ~ +prod_amount_y + population,
    data = normalized)

Residuals:
     Min       1Q   Median       3Q      Max
-1.12455 -0.18443 -0.01338  0.20883  0.76833

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.481e-17  5.293e-02    0.00        1
prod_amount_y  6.220e-01  5.350e-02   11.63 2.73e-15 ***
population     6.738e-01  5.350e-02   12.59  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3705 on 46 degrees of freedom
Multiple R-squared:  0.8685,	Adjusted R-squared:  0.8627
F-statistic: 151.8 on 2 and 46 DF,  p-value: < 2.2e-16
```

Both are highly significant but R^2 is getting lower. Findings support OECD but there could be more factors

Remove highly correlated variables

```
# Discovering highly correlated explanatory
  variables
hicorvars = findCorrelation(cor(feats), cutoff =
  0.7)
expvarsnohc = paste(colnames(feats[,-hicorvars]),
  collapse = "+")
formulanohc = paste(target,"~",expvarsnohc,collapse
  = "+")
mod_varnohc = summary(lm(paste(target, '~',
  expvarsnohc) ,data = normalized))
```

Highly correlated variables:

gni_pc, agri_gdp, population, daily_caloric_supply, exp_veg, exp_sug, avg_p_barrel

Remove highly correlated variables

This supports the OECD's statement but we throw away features known to be significant. However the model still has a higher adjusted R^2 than small model, imp_veg is correlated with population by almost 0.70 and could be used interchangeably

```
Call:
lm(formula = formulanohc, data = normalized)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.68406 -0.23264  0.03568  0.18202  0.65686 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -5.169e-15  4.954e-02   0.000  1.00000    
pr_q1         -5.590e-01  3.196e-01  -1.749  0.08929 .  
pr_q2         -1.245e-01  1.004e-01  -1.241  0.22319    
pr_q3         -2.154e-01  2.367e-01  -0.910  0.36923    
pr_q4          2.535e-01  2.278e-01   1.113  0.27358    
tas_q1        -5.202e-01  3.244e-01  -1.604  0.11806    
tas_q2         2.817e-01  2.060e-01   1.367  0.18060    
tas_q3        -5.288e-01  5.904e-01  -0.896  0.37670    
tas_q4         1.720e-01  1.225e-01   1.404  0.16933    
prod_amount_y  6.338e-01  5.052e-02  12.546  2.6e-14 ***
exp_cer       -3.440e-01  2.343e-01  -1.468  0.15117    
imp_sug        4.485e-01  6.668e-01   0.673  0.50574    
imp_veg        1.207e+00  3.646e-01   3.310  0.00221 ** 
imp_cer        5.652e-01  4.027e-01   1.404  0.16949    
cp_inflation  -3.120e-01  3.709e-01  -0.841  0.40616    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3468 on 34 degrees of freedom
Multiple R-squared:  0.9148,	Adjusted R-squared:  0.8797 
F-statistic: 26.08 on 14 and 34 DF,  p-value: 3.992e-14
```

Stepwise approach to tackle multicolinearity

- Now we want to keep some of the variables we didn't use previously. We make our selection based on variance inflation factor (VIF)
- Multicorrelated variables increase the variance of the model.
- The degree of variance added by a variable can be measured by the VIF:

Stepwise approach to tackle multicolinearity

1. Given that VIF has been calculated initially for each variable:
2. **while** *(max(VIF) >= cutoffvalue)* **do**

    Remove variable with highest VIF

    for *(1:Number of remaining variables)* **do**

        take x_i as target

        the others as predictors

        Calculate coefficient of determination R^2 with regression

$$x_i = a_2 x_2 + a_3 x_3 + \ldots + a_k x_k + c_0 + e \tag{1}$$

$$VIF_i = \frac{1}{1 - R^2} \tag{2}$$

    end

end

Function to remove MC

```
removeVif<-function(explan_vars,cutoffval=10){
  if(!require("fmsb")) install.packages("fmsb"); library("fmsb")
  tempresults = as.data.frame(matrix(ncol = 2, nrow = 0))
  colnames(tempresults) = c("variable","vif")
  #initially calculate VIF for each explanatory variable
  for (i in 1:NROW(colnames(explan_vars)) ){
    temptarget = colnames(explan_vars)[i]
    tempexpvars = paste(colnames(explan_vars[,!(colnames(explan_vars) %in% temptarget)]), collapse = "+")
    tempformula = paste(temptarget,"~", tempexpvars, collapse = " ")
```

```
    tempresults[i,1] = temptarget
    tempresults[i,2] = VIF(lm( tempformula,data =
      explan_vars))
  }
  print(tempresults[order(tempresults$vif),])
  #remove variable with highest VIF, calculate new
    VIF for remaining variables until all VIF are
    below cutoff value
  while(max(tempresults$vif) >= cutoffval){
    tempresults = tempresults[!tempresults$vif ==
      max(tempresults$vif),]
    tempremvars = tempresults$variable
    for(j in 1: NROW(tempremvars)){
      temptarget = tempremvars[j]
      tempexpvars = paste(tempremvars[!tempremvars %
        in% temptarget],collapse = "+")
      tempformula = paste(temptarget,"~",
        tempexpvars, collapse = " ")
```

```
      tempresults[j,1] = temptarget
      tempresults[j,2] = VIF(lm( tempformula,data =
        explan_vars))
    }
    print("Remaining variables:")
    print(tempresults[order(tempresults$vif),])
    cat("\n")
  }
  return(tempresults$variable)
}
```

Combined approach

We apply the removeVif function to the highly correlated variables:

```
# for highly correlated variables
varslovifhc = removeVif(feats[,hicorvars],8)
```

```
[1] "Remaining variables:"
             variable      vif
6             exp_sug 3.391032
7         avg_p_barrel 3.450679
3          population 4.954443
4 daily_caloric_supply 5.983362
```

Combined approach

Apply the same function to the other variables and build the mode:

```
varslovifnohc = removeVif(feats[,-hicorvars],8)
# Model without multicolinearity
expvars_lovif = paste(paste(varslovifhc,collapse = "
  +"),"+",paste(varslovifnohc,collapse = "+"),
  collapse = "+")
formula_lovif = paste(target,"~",expvars_lovif,
  collapse = "+")
mod_varnohc = summary(lm(formula_lovif,data =
  normalized))
```

Combined approach

The results also support the oecd's statement, results are similar to previous approach

```
Call:
lm(formula = formula_lovif, data = normalized)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66128 -0.22110  0.02984  0.20824  0.67537

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.051e-15  4.994e-02   0.000   1.0000
population           1.400e+00  6.416e-01   2.182   0.0364 *
daily_caloric_supply -5.156e-01  1.007e+00  -0.512   0.6121
exp_sug              2.476e+00  2.538e+00   0.976   0.3364
avg_p_barrel        -3.768e-01  8.388e-01  -0.449   0.6562
pr_q1               -2.734e-01  1.687e-01  -1.620   0.1147
pr_q2               -5.552e-01  5.523e-01  -1.005   0.3221
pr_q3               -1.205e+00  1.046e+00  -1.152   0.2575
pr_q4                1.353e+00  1.115e+00   1.213   0.2337
tas_q2              -2.587e-01  5.909e-01  -0.438   0.6644
tas_q3               3.414e-02  2.896e-01   0.118   0.9069
tas_q4              -1.610e-01  2.397e-01  -0.672   0.5064
prod_amount_y        6.330e-01  5.094e-02  12.425 5.42e-14 ***
exp_cer             -9.763e-01  8.713e-01  -1.121   0.2706
imp_cer              9.164e-01  8.980e-01   1.020   0.3149
cp_inflation        -2.274e-01  6.704e-01  -0.339   0.7366
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3496 on 33 degrees of freedom
Multiple R-squared:  0.916,	Adjusted R-squared:  0.8778
F-statistic: 23.99 on 15 and 33 DF,  p-value: 1.684e-13
```

Results

Model	Significant Variables	R^2
Naive approach	prod_amount_y	0.8778
Based on correlation significance	prod_amount_y	0.8753
Small model	prod_amount_y population	0.8627
After removing highly correlated variables	prod_amount_y imp_veg pr_q1	0.8797
Combined approach	prod_amount_y population	0.8778

Related Work

Vasilii Erokhin
Factors Influencing Food Markets in Developing Countries: An Approach to Assess Sustainability of the Food Supply in Russia
available on www.mdpi.com, 2017

Smith, M.E.
Smith, M.E. World Food Security. The Effect of U.S. Farm Policy; United States Department of Agriculture: Washington, WA, USA, 1990. : Distinction between supply and demand related factors

Outlook

What's left to do:

- Integrate more production related data e.g. stock, production costs in agriculture
- Increase observation period
- Enable analysis of single products
- Compare datasets of other countries
- Compare other feature selection techniques, e.g. random forest based importance.