Is the World Bank Financing Bad Governance in Developing Countries?

Raulin L. Cadet

4 octobre 2019

In developing countries with high level of corruption, governance become worst when they receive grants from the World Bank.

# The data are imported bellow. The first one have data about grants fom the World. The second have the index of control for corruption. Higher index of control for corruption is a better performance.  
grants<-read.csv("https://finances.worldbank.org/api/views/34vv-5t6y/rows.csv?accessType=DOWNLOAD&bom=true&format=true")  
  
sheetName="GovernmentEffectiveness"  
corrup<-read.xlsx("wgidataset.xlsx",sheetName ="GovernmentEffectiveness" ) #"ControlofCorruption" Source: https://info.worldbank.org/governance/wgi/Home/downLoadFile?fileName=wgidataset.xlsx

grant<-data.frame(grants$Country,grants$Region,grants$Original.Principal.Amount,grants$Agreement.Signing.Date)  
colnames(grant)<-c("Country","Region","Amount","SignedDate")  
  
# Function to convert date to Year only  
dateY<-function(x,formatDate){   
 d=as.Date(x,format = formatDate)  
 new\_d=as.numeric(format(d,"%Y"))  
 return(new\_d)  
}  
  
# Creating a column for the year of signed agreements between the World Bank and the beneficiaries of grants.  
grant['Year']=dateY(grant$SignedDate,format="%m/%d/%Y")  
  
str(grant) # shows that the variable Amount is a factor. It should be numeric.

## 'data.frame': 957 obs. of 5 variables:  
## $ Country : Factor w/ 76 levels "Afghanistan",..: 71 49 31 10 18 1 1 1 1 18 ...  
## $ Region : Factor w/ 6 levels "AFRICA","EAST ASIA AND PACIFIC",..: 1 1 4 1 1 6 6 6 6 1 ...  
## $ Amount : Factor w/ 291 levels "1,000,000.00",..: 264 62 173 241 209 10 163 53 186 190 ...  
## $ SignedDate: Factor w/ 679 levels "","01/04/2011",..: 157 669 640 672 448 137 260 260 260 453 ...  
## $ Year : num 1998 1998 2000 2000 2001 ...

grant$Amount=as.numeric(grant$Amount) # converting the variable "Amount" as numeric

# Calculate total of grant, by country, by year  
grant2<-grant%>%  
 group\_by(Country,Region,Year)%>%  
 summarize(Total=sum(Amount))  
  
head(grant2,4) # Viewing the first four rows of the data frame 'grant2'

## # A tibble: 4 x 4  
## # Groups: Country, Region [1]  
## Country Region Year Total  
## <fct> <fct> <dbl> <dbl>  
## 1 Afghanistan SOUTH ASIA 2002 412  
## 2 Afghanistan SOUTH ASIA 2003 574  
## 3 Afghanistan SOUTH ASIA 2004 457  
## 4 Afghanistan SOUTH ASIA 2005 915

# The variable is control of corruption. Then high value is better  
colnames(corrup)<-t(corrup[12,]) # Add names of columns, available in line 12.  
corrup2<-corrup[13:226,] # Selecting data starting on row 12. The rows above 12 do not contain data, but title information.  
  
# Selecting only the estimate's columns and the country name column. They provide the index data. The other columns are other statistics about the index.  
corrup3=data.frame(corrup2$`Country/Territory`)  
for(i in 1:dim(corrup2)[2]){  
 if(colnames(corrup2[i])=="Estimate"){  
 corrup3[as.character(i)]=corrup2[,i]  
 }  
   
}  
  
# Names the columns of the new data frame with the years  
Year<-unique(c(t(corrup[11,])))[-1]  
Year

## [1] "1996" "1998" "2000" "2002" "2003" "2004" "2005" "2006" "2007" "2008"  
## [11] "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017" "2018"

colnames(corrup3)<-c("Country",Year)  
  
# Selecting data from year 2002  
year<-seq(2002,2015,1)  
corrup3<-corrup3[,c("Country",year)]  
  
######## Select in corrup only countries that receive grants #####  
corrupT<-t(corrup3) # Transpose the dataframe to have the countries name in columns  
colnames(corrupT)<-t(corrupT[1,]) # Change default name to countries name which are in row 1.  
corrupT<-corrupT[-1,] # Remove the first row that contains the countries names  
  
unq<-function(x){ # Function that returning a vector of unique values separated by comma  
 y=NULL  
 for(i in 1:length(x)){  
 if(!x[i] %in% y){  
 y=c(y,x[i])  
 }  
 }  
 return(y)  
}  
  
  
grant\_countries<-unique(as.character(grant2$Country)) # Names of countries that receive grants from 2002 to 2015  
  
  
###########################################################################################  
######## Correct difference in countries names spelling, between the dataframe ############  
cog=NULL  
for(i in 1:length(grant\_countries)){ # Identify difference in countries names in the two data frame  
if(!grant\_countries[i] %in% colnames(corrupT)){  
 cog=c(cog,grant\_countries[i])  
 }  
}  
  
# Data frame that contains names spelling that are differents  
CoSpell<-data.frame(corName=c("Central African Republic", "Congo, Dem. Rep.","Congo, Rep.","Lao PDR","Micronesia, Fed. Sts.","SÃ£o TomÃ© and Principe","Yemen, Rep.","CÃ´te d'Ivoire"),  
 grantName=c("Central Africa","Congo, Democratic Republic of", "Congo, Republic of","Lao People's Democratic Republic","Micronesia, Federated States of","Sao Tome and Principe","Yemen, Republic of","Cote d'Ivoire"))  
  
  
for(i in 1:dim(CoSpell)[1]){ # Correct the names in the data frame of control for corruption  
 cf=gsub(CoSpell$corName[i],CoSpell$grantName[i],colnames(corrupT))  
}  
  
colnames(corrupT)<-cf # add corrected names as column names of the data frame of control for corruption  
  
##########################################################################################  
# Identify countries names that appear in both data frame  
co=NULL  
for(i in 1:length(grant\_countries)){   
if(grant\_countries[i] %in% colnames(corrupT)){  
 co=c(co,grant\_countries[i])  
 }  
}  
  
corrupFi<-corrupT[,co];dim(corrupFi) # This data frame (corrupFi) has 62 countries found in grant2

## [1] 14 62

corrupFinal<-data.frame(newcol = c(t(corrupFi)), stringsAsFactors=FALSE) # Create a data frame of one column, with data of control of corruption for all countries that receive grants from the WB.  
  
  
####### continue below  
  
  
  
  
  
  
corrupFinal["Year"]<-rep(year,dim(corrupFinal)[2]) # Adding the column of the year to the new data frame.  
  
  
rep2<-function(x,n){ # This function should repeat each element of a vector n times   
 y=NULL  
 for(i in 1:length(x)){  
 y=c(y,rep(x[i],n))  
 }  
 return(y)  
}  
  
  
corrupFinal["Country"]<-rep2(co,length(year));dim(corrupFinal) # Data frame with index of control for corruption, for countries that receive grant from the World Bank.

## [1] 868 3

# Data frame of grants with only country found in the data frame 'corrupFinal'. However, data are not available for some years, for some countries. That's why they do not have the same number of rows.   
Grant<-grant2 %>%   
 filter(Country %in% co)  
dim(Grant)

## [1] 352 4

# Merging data frame 'Grant' and 'corrupFinal' where country name and year are the same  
DatFinal<-inner\_join(Grant, corrupFinal, by = NULL, copy = FALSE, suffix = c(".grant", ".corrup") )

## Joining, by = c("Country", "Year")

## Warning: Column `Country` joining factor and character vector, coercing  
## into character vector

colnames(DatFinal)<-c(colnames(DatFinal)[1:3],"Grant","Control")

str(DatFinal)

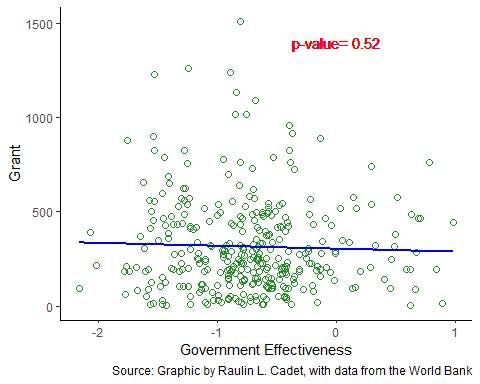
## Classes 'grouped\_df', 'tbl\_df', 'tbl' and 'data.frame': 347 obs. of 5 variables:  
## $ Country: chr "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ Region : Factor w/ 6 levels "AFRICA","EAST ASIA AND PACIFIC",..: 6 6 6 6 6 6 6 6 6 6 ...  
## $ Year : num 2002 2003 2004 2005 2006 ...  
## $ Grant : num 412 574 457 915 760 ...  
## $ Control: chr "-1.56797313690186" "-1.23885667324066" "-0.688740909099579" "-0.367727726697922" ...  
## - attr(\*, "groups")=Classes 'tbl\_df', 'tbl' and 'data.frame': 61 obs. of 3 variables:  
## ..$ Country: chr "Afghanistan" "Angola" "Bangladesh" "Benin" ...  
## ..$ Region : Factor w/ 6 levels "AFRICA","EAST ASIA AND PACIFIC",..: 6 1 6 1 6 1 1 2 1 1 ...  
## ..$ .rows :List of 61  
## .. ..$ : int 1 2 3 4 5 6 7 8 9 10 ...  
## .. ..$ : int 15 16  
## .. ..$ : int 17 18  
## .. ..$ : int 19 20 21 22 23 24  
## .. ..$ : int 25 26 27 28 29  
## .. ..$ : int 30 31 32 33 34 35 36 37 38 39 ...  
## .. ..$ : int 42 43 44 45 46 47 48 49 50 51 ...  
## .. ..$ : int 53 54 55 56 57 58 59  
## .. ..$ : int 60  
## .. ..$ : int 61 62 63 64 65 66 67 68  
## .. ..$ : int 69 70 71 72 73 74 75 76  
## .. ..$ : int 77 78 79 80 81 82  
## .. ..$ : int 83 84 85 86 87 88 89  
## .. ..$ : int 90 91 92 93 94 95 96 97  
## .. ..$ : int 98 99 100  
## .. ..$ : int 101 102 103 104 105 106 107 108 109  
## .. ..$ : int 110 111 112 113 114 115 116 117  
## .. ..$ : int 118 119  
## .. ..$ : int 120 121 122 123 124  
## .. ..$ : int 125 126 127 128 129 130 131 132  
## .. ..$ : int 133 134 135 136 137 138 139  
## .. ..$ : int 140 141 142  
## .. ..$ : int 143 144 145 146 147 148 149 150 151 152  
## .. ..$ : int 153 154  
## .. ..$ : int 155 156 157  
## .. ..$ : int 158 159 160 161 162  
## .. ..$ : int 163 164 165 166 167 168 169 170 171 172 ...  
## .. ..$ : int 176 177 178 179 180 181 182 183 184  
## .. ..$ : int 185 186 187 188 189 190  
## .. ..$ : int 191  
## .. ..$ : int 192 193 194 195 196 197 198 199 200 201 ...  
## .. ..$ : int 204 205 206 207  
## .. ..$ : int 208 209 210 211 212 213  
## .. ..$ : int 214  
## .. ..$ : int 215 216 217 218  
## .. ..$ : int 219 220 221  
## .. ..$ : int 222 223 224  
## .. ..$ : int 225 226 227 228  
## .. ..$ : int 229  
## .. ..$ : int 230 231 232 233 234 235 236 237 238 239 ...  
## .. ..$ : int 241 242 243 244 245 246 247  
## .. ..$ : int 248 249 250 251 252 253 254 255  
## .. ..$ : int 256  
## .. ..$ : int 257 258 259 260 261 262 263 264 265 266 ...  
## .. ..$ : int 269 270 271 272  
## .. ..$ : int 273 274 275 276 277 278 279 280 281 282 ...  
## .. ..$ : int 285 286 287 288 289 290 291  
## .. ..$ : int 292  
## .. ..$ : int 293 294 295 296  
## .. ..$ : int 297  
## .. ..$ : int 298  
## .. ..$ : int 299 300 301 302 303 304 305 306 307 308 ...  
## .. ..$ : int 312 313 314  
## .. ..$ : int 315 316 317 318 319 320 321 322  
## .. ..$ : int 323 324 325 326 327 328 329  
## .. ..$ : int 330 331 332 333 334 335  
## .. ..$ : int 336 337 338  
## .. ..$ : int 339 340 341 342  
## .. ..$ : int 343  
## .. ..$ : int 344  
## .. ..$ : int 345 346 347  
## ..- attr(\*, ".drop")= logi TRUE

DatFinal <- within(DatFinal, { # Changing type of some variable in 'DatFinal'   
 Country <- factor(Country) # Convert 'Country' to factor  
 Control<-as.numeric(Control) # Convert 'Control' to numeric  
 })

## Warning in eval(substitute(expr), e): NAs introduits lors de la conversion  
## automatique

A scatter plot of the linkage between grants and government effecctiveness is presented below. It shows that the World Bank tends not to account for how the population perceive their governments effectiveness to grant funds to their countries.

#cor(DatFinal$Grant,DatFinal$Control)  
gg1 <- ggplot(DatFinal, aes(x=Control, y=Grant)) +   
 geom\_point(col='forestgreen',size=2,shape=1) +   
 geom\_smooth(method = 'lm', se=F,col='blue') +   
 #facet\_wrap(~Region) +  
 #xlim(c(0, 0.1)) +   
 #ylim(c(0, 500000)) +   
 labs(#subtitle="Area Vs Population",   
 y="Grant",   
 x="Government Effectiveness",   
 # title="Linkage between Grant from the World Bank and Governement Effectiveness",   
 caption = "Source: Graphic by Raulin L. Cadet, with data from the World Bank")+  
 theme\_classic()  
  
gg1+geom\_text(aes(x=0,y=1400,label=paste("p-value=",round(cor.test(Grant,Control)$p.value,digits = 2))),col='red')



Considering a scotter plot, for the same variables, but by region, only the graphic related to Africa reveals that grants tend to increase with high perception of government effectiveness. However, this relation seems to bo weak. Considering East Asia and Pacific, the curve estimating the raltion between grants and government effectiveness is almost flat, showing no relation between them. For the other regions, the relation is not consistent, suggesting that the World Bank is not following a rule related to government effectiveness to grant fund.

The absence of such a rule sould not encourage government of developing countries to improve the effectiveness of public policy.

Let us verify if there is a significant linear relationship between the grants and government effectiveness.

cor<-DatFinal%>% # Calculating the correlation coefficient (method of Pearson)  
 group\_by(Region)%>% # Grouping by region  
 summarise(cor.test(Control,Grant)$estimate)  
  
pVal<-DatFinal%>% # Calculating the p-value of the correlation by region  
 group\_by(Region)%>%  
 summarise(cor.test(Control,Grant)$p.value)

# Merging the data frame 'DatFInal' with data related to p-value for correlation test by region  
# to include p-value in the following graphic  
datfinal<-inner\_join(DatFinal, pVal, by = NULL, copy = FALSE, suffix = c(".grant", ".corrup") )  
colnames(datfinal)<-c(colnames(DatFinal)[1:5],"P\_value")  
  
  
gg2 <- ggplot(datfinal, aes(x=Control, y=Grant)) + # Scatter plot linking 'Grant' and 'Control'  
 geom\_point(col='forestgreen',size=2,shape=1) +   
 geom\_smooth(method = 'lm', se=F,col='blue') + # Design the linear relationship (blue line)  
   
 facet\_wrap(~Region)+ # Design a graphic by region  
 labs(subtitle="By Region (2002-2015)",   
 y="Grant",   
 x="Government Effectiveness",   
 title="Linkage between Grant from the World Bank and Governement Effectiveness",   
 caption = "Source: Graphic by Raulin L. Cadet, with data from the World Bank")+  
 theme\_classic()  
gg2+geom\_text(aes(x=-1,y=1200,label=paste("p-value=",round(P\_value,digits = 2))),col='red')

