Introduction A l’Informatique pour Sciences Sociales\_Devoir Final\_UNIQ

Djim C. Guerrier Code: GU140127

2023-03-25

# 1. Courte introduction expliquant pourquoi il est important d’identifier les déterminants de l’indicateur d’intérêt

La variable d’interet est l’Exportation de biens et de services (En Dollar US courant).

Il est important d’identifier les déterminants de l’exportation d’un pays car cela peut aider les gouvernements et les entreprises à prendre des décisions éclairées sur les politiques économiques et commerciales. En comprenant les facteurs qui affectent les exportations, les décideurs peuvent élaborer des politiques visant à améliorer la compétitivité des entreprises exportatrices, à renforcer les relations commerciales avec les partenaires commerciaux clés, à diversifier les exportations et à élargir leurs marchés d’exportation.

De plus, les exportations peuvent avoir un impact important sur la croissance économique et le développement d’un pays, surtout pour les économies en développement qui dépendent souvent des exportations pour accéder aux marchés internationaux afin d’accroître leurs revenus. Identifier les déterminants des exportations peut donc aider à stimuler la croissance économique et la création d’emplois dans les pays exportateurs.

library(wbstats)

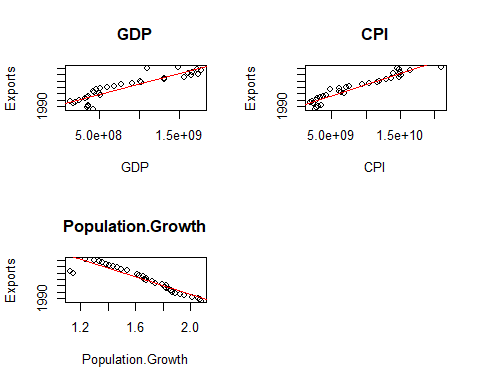
HT\_data <- wbstats::wb\_data(indicator = c("NE.EXP.GNFS.CD", "NY.GDP.MKTP.CD", "FP.CPI.TOTL.ZG", "SP.POP.GROW"),country = "Haiti", start = 1988, end = 2022)  
   
  
#(WDI(country = "HT", indicator = c("NE.EXP.GNFS.CD", "NY.GDP.MKTP.CD", "FP.CPI.TOTL.ZG", "SP.POP.GROW"), start = 1988, end = 2022))  
  
names(HT\_data)=c("Haiti","HT","HTI","Year","Exports","GDP","CPI","Population.Growth")  
Frame\_HT\_Data <- data.frame(HT\_data[,-c(1:3)]) #Ici on porte les donnees des variables dans un dataframe puis on enleve les 3 premieres colonnes puisqu'elles contiennent des informations non-importantes  
Frame\_HT\_Data

## Year Exports GDP CPI Population.Growth  
## 1 1988 4.105468 426125800 2613926800 2.077271  
## 2 1989 6.923905 356133600 2736243800 2.073211  
## 3 1990 21.276032 368773800 3096289800 2.059522  
## 4 1991 15.419907 358172245 3473540602 2.011648  
## 5 1992 19.358495 170898224 2257121668 1.958115  
## 6 1993 29.705972 203702653 1878248741 1.926430  
## 7 1994 39.331620 139173370 2167564195 1.889899  
## 8 1995 27.608185 256955765 2813313279 1.868925  
## 9 1996 20.583596 329352514 2907514523 1.861047  
## 10 1997 20.559007 349003994 3338938830 1.843494  
## 11 1998 5.269035 368684570 3723909227 1.824329  
## 12 1999 3.004394 508795816 4153736347 1.819921  
## 13 2000 9.333222 500598000 6813577558 1.826141  
## 14 2001 13.316722 445213583 6331961555 1.795964  
## 15 2002 7.032874 421180249 6058134467 1.744825  
## 16 2003 28.699578 469754619 4826827577 1.724900  
## 17 2004 21.031834 518490700 6036959915 1.679416  
## 18 2005 13.973008 605069775 7184064657 1.664486  
## 19 2006 11.345215 689103494 7518107787 1.680162  
## 20 2007 6.557228 779201474 9522763154 1.653990  
## 21 2008 15.281774 917202249 10485225353 1.625856  
## 22 2009 0.393886 1034003272 11597014807 1.609813  
## 23 2010 4.827342 1016377476 11859315079 1.146889  
## 24 2011 6.332795 1323610001 13008754111 1.125747  
## 25 2012 5.018042 1323609690 13708926466 1.537469  
## 26 2013 4.765024 1566906102 14902474091 1.498986  
## 27 2014 3.438167 1662116687 15139264670 1.465968  
## 28 2015 6.731841 1747794244 14833154472 1.439894  
## 29 2016 11.502907 1613347566 13987693739 1.410821  
## 30 2017 10.680089 1664384136 15035560373 1.387530  
## 31 2018 12.481411 1779807736 16455034353 1.361131  
## 32 2019 18.703783 1732030797 14785839383 1.335139  
## 33 2020 22.796311 1110024755 14508218017 1.302920  
## 34 2021 16.841524 1490135681 20944392615 1.237299

View(Frame\_HT\_Data) #Afin de verifier que tout est en ordre

# Charger le package pour réaliser la régression linéaire  
library(stats)

# Création du graphe pour chaque variable indépendante  
par(mfrow = c(2, 2)) # Diviser la zone de graphique en 4 parties  
for(i in 3:5) {  
 plot(Frame\_HT\_Data[, i], Frame\_HT\_Data[, 1], main = colnames(Frame\_HT\_Data)[i], xlab = colnames(Frame\_HT\_Data)[i], ylab = "Exports")  
 abline(lm(Frame\_HT\_Data[, 1] ~ Frame\_HT\_Data[, i]), col = "red")  
}   
#Dans le code precedent, on a utilise une boucle afin d'eviter a avoir a ecrire la meme ligne pour chaque variable individuellement



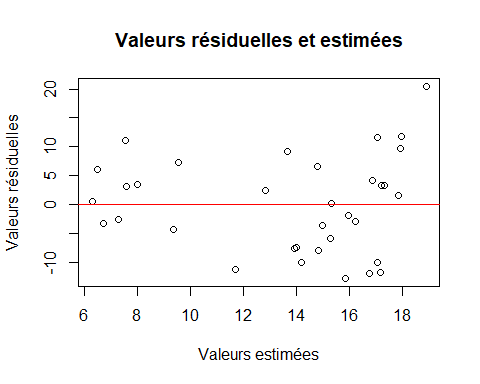
# Régression de la variable dépendante sur les variables indépendantes  
model=lm(Exports~GDP+CPI+Population.Growth, data = Frame\_HT\_Data)  
summary(model)

##   
## Call:  
## lm(formula = Exports ~ GDP + CPI + Population.Growth, data = Frame\_HT\_Data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.8210 -7.0652 0.2715 5.5281 20.4542   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 4.070e+01 2.618e+01 1.555 0.130  
## GDP -9.149e-09 8.760e-09 -1.044 0.305  
## CPI -2.087e-10 1.098e-09 -0.190 0.850  
## Population.Growth -1.064e+01 1.276e+01 -0.834 0.411  
##   
## Residual standard error: 8.648 on 30 degrees of freedom  
## Multiple R-squared: 0.1902, Adjusted R-squared: 0.1092   
## F-statistic: 2.349 on 3 and 30 DF, p-value: 0.09238

# Création d'un dataframe pour les résultats de la régression du modele  
summary\_df <- data.frame(coefficients = coef(model), p\_values = summary(model)$coefficients[, 4])  
  
summary\_df$coefficients[1] <- NA # On retire le coefficient pour la constante  
summary\_df$statistic <- c(summary(model)$r.squared, summary(model)$fstatistic[1])  
colnames(summary\_df)[2] <- "Fisher"  
summary\_df$p\_values[4] <- NA # On retire le p-value pour la statistique de Fisher  
summary\_df$R2\_adj <- summary(model)$adj.r.squared  
rownames(summary\_df)[1] <- "Exports"  
rownames(summary\_df)[2] <- "GDP"  
rownames(summary\_df)[3] <- "CPI"  
rownames(summary\_df)[4] <- "Population.Growth"  
  
summary\_df

## coefficients Fisher statistic p\_values R2\_adj  
## Exports NA 0.1304367 0.1902195 NA 0.1092415  
## GDP -9.149339e-09 0.3046465 2.3490259 NA 0.1092415  
## CPI -2.086968e-10 0.8504792 0.1902195 NA 0.1092415  
## Population.Growth -1.063532e+01 0.4111177 2.3490259 NA 0.1092415

# Création d'un graphique pour les valeurs résiduelles et les valeurs estimées  
  
{plot(model$fitted.values, model$residuals, main = "Valeurs résiduelles et estimées", xlab = "Valeurs estimées", ylab = "Valeurs résiduelles")  
abline(h = 0, col = "red")}



```

# 2. Commentaire sur le graphique en nuage de points, mettant en lien les valeurs résiduelles (erreurs) de la régression et les valeurs estimées (fitted values)

dans le graphique,le schema formé par la dispersion des points accuse d’une répartition plus ou moins équilibrée des valeurs résiduelle en fonction des valeurs estimées des deux coté de la ligne de zéro (sauf une seule valeur aberrante- en haut a droite), et ne présente aucune forme identifiable. Ces observations nous indiquent que le modele est plus ou moins solide et satisfait pas les hypotheses de loi normale et de la constance de la variance des erreurs.