AQLI Test Packet - EPIC

1. Data Management, Cleaning and Analysis Section

Priyanka Gagneja

2023-11-24

## AQLI @ EPIC

#### Initial Set up

library(magrittr)  
library(ggplot2)  
library(ggtext)  
library(patchwork) # for modifying plot output placements  
library(easystats) # group of pkgs for regression modelling and diagnostiics  
library(plm) # for panel data

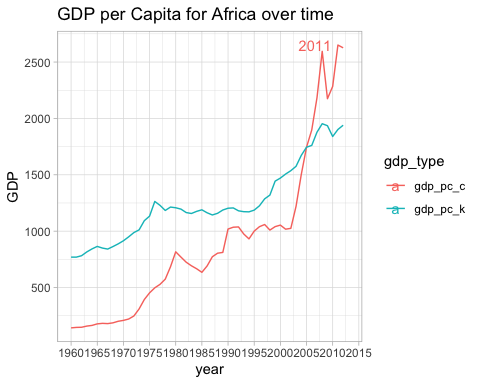
### 1.1 Data Cleaning

# (1) Load the data  
Africa\_gdp <- haven::read\_dta("Inputs/task1/Africa\_gdp.dta")  
Africa\_countries\_codes <- readxl::read\_excel("Inputs/task1/Africa\_countries\_codes.xlsx")  
  
Africa\_gdp <- Africa\_gdp %>%   
   
 # (2) Bring in the variable WB2code. Also assign a numeric code using variable name NumCode to each country, with each number reflecting the rank of the country in alphabetical order.  
 dplyr::left\_join(Africa\_countries\_codes %>%   
 dplyr::arrange(CountryName) %>%   
 dplyr::mutate(NumCode = rank(CountryName)) %>%   
 dplyr::select(-CountryName),   
 by = "CountryCode") %>%   
   
 # (3) Create LName cleaned version of LongName  
 dplyr::mutate(LName = gsub(paste(c("of","the"), collapse = "|"), "", LongName))   
   
 # (4)(a) Generate a table with averages over time of gdp\_pc\_k and gdp\_pc\_c for every country.   
average\_gdp <- Africa\_gdp %>%   
 dplyr::group\_by(CountryCode) %>%   
 dplyr::summarise(gdp\_pc\_k = mean(gdp\_pc\_k, na.rm = TRUE),  
 gdp\_pc\_c = mean(gdp\_pc\_c, na.rm = TRUE)) %>%   
 dplyr::ungroup()  
   
 # (4)(b) Export the table in publishable format as an excel file and name it “average\_gdp.xlsx” (or “.csv”)  
 write.csv(average\_gdp, "Outputs/task1/average\_gdp.csv")  
  
 # (5)(a) Load the “WDI\_Agriculture\_VA.csv”  
 WDI\_Agriculture\_VA <- read.csv("Inputs/task1/WDI\_Agricultural\_VA.csv") %>%   
   
 # (5)(b) Change the format of data from wide to date wise long format,   
 # with Year on one column and Agriculture constant values on the other  
 dplyr::select(CountryCode = Country.Code, contains("X"), -X2.alpha.code) %>%   
 tidyr::pivot\_longer(cols = contains("X"), names\_to = "Year", values\_to = "Agri\_Const") %>%   
 dplyr::mutate(Year = sub("X","",Year))  
  
 # (6)(a) Merge the WDI\_Agriculture\_VA dataset with the Africa\_gdp dataset by Year and Country  
 Africa\_merged <- WDI\_Agriculture\_VA %>%   
 dplyr::right\_join(Africa\_gdp %>%   
 dplyr::mutate(Year = as.character(year)) %>%   
 dplyr::select(-year),   
 by = c("CountryCode", "Year"))   
   
 # (6)(b) Export the dataset and name it “Africa\_merged.xlsx” (or .csv)  
 write.csv(Africa\_merged, "Outputs/task1/Africa\_merged.csv")  
  
   
 # (7)(a) Generate a separate timeseries dataset which depicts the average constant and current GDP per capita (gdp\_pc\_k and gdp\_pc\_c) of Africa from 1960 to 2013. Export the table in excel format and name it “Africa\_gdp\_timeseries.xlsx” (or .csv)  
   
 Africa\_gdp\_timeseries <- Africa\_gdp %>%   
 dplyr::group\_by(year) %>%   
 dplyr::summarise(gdp\_pc\_k = mean(gdp\_pc\_k, na.rm = TRUE),  
 gdp\_pc\_c = mean(gdp\_pc\_c, na.rm = TRUE)) %>%   
 dplyr::ungroup()  
   
 # (7)(b) Export the table in publishable format as an excel file and name it “average\_gdp.xlsx” (or “.csv”)  
 write.csv(Africa\_gdp\_timeseries, "Outputs/task1/Africa\_gdp\_timeseries.csv")  
   
   
   
 # (8)(a) Generate a separate data set of the following variables average by country: i) GDP per capita (current), ii) GDPpercapita(constant), iii) Agriculture value (constant)  
 Africa\_country\_average <- Africa\_merged %>%   
 dplyr::group\_by(Region, CountryName) %>%   
 dplyr::summarise(gdp\_pc\_k = mean(gdp\_pc\_k, na.rm = TRUE),  
 gdp\_pc\_c = mean(gdp\_pc\_c, na.rm = TRUE),  
 Agri\_Const = mean(Agri\_Const, na.rm = TRUE)) %>%   
 dplyr::ungroup()  
  
 # (8)(b) Save and export the dataset as “Africa\_country\_average.xlsx” (or .csv). Make sure to retain the region variable.  
 write.csv(Africa\_country\_average %>%   
 dplyr::rename(Country = CountryName,  
 "GDP per capita (current)" = gdp\_pc\_k,  
 "GDP per capita (constant)" = gdp\_pc\_c,  
 "Agriculture value (constant)" = Agri\_Const),   
 "Outputs/task1/Africa\_country\_average.csv")

### 1.2 Data Exploration

1. Using the Africa\_gdp\_timeseries dataset, generate line graph of average constant and current GDP per capita against time for Africa. On a particular year, the average GDP per capita seem to increase drastically. On what year does there seem to be a drastic increase? Indicate the year on the plot. Format your graphs so that they are publication quality and save them as .pdf files.

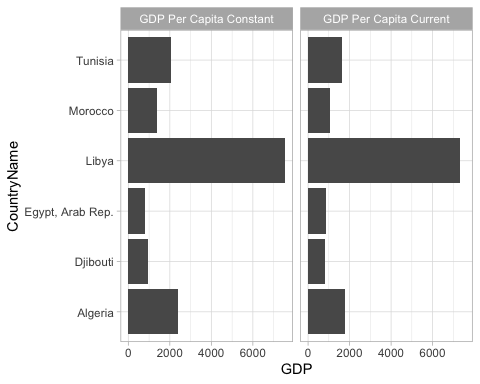
Africa\_gdp\_ts <- Africa\_gdp\_timeseries %>%   
 tidyr::pivot\_longer(cols = c(gdp\_pc\_k, gdp\_pc\_c), names\_to = "gdp\_type", values\_to = "GDP")  
  
# avg\_ts\_plot <- Africa\_gdp\_timeseries %>%   
Africa\_gdp\_ts %>%   
 ggplot(aes(x = year, y = GDP, colour = gdp\_type)) +   
 geom\_line() +   
 # facet\_wrap(~gdp\_type) +   
 geom\_text(aes(label = year), data = Africa\_gdp\_ts[which.max(Africa\_gdp\_ts$GDP),], hjust = 1.2, vjust = 0.5) +  
 # ggtext::geom\_textbox(aes(label = paste0("<span style='font-size:32pt'><br>", Africa\_gdp\_timeseries[which.max(Africa\_gdp\_timeseries$GDP),], "<br></span>")), vjust = 0.45, fill = NA, box.colour = NA, family = "Cabin", size = 7, fontface = "bold") +   
 scale\_x\_continuous(n.breaks = 10) +   
 scale\_y\_continuous(n.breaks = 6) +   
 labs(title = "GDP per Capita for Africa over time") +   
 theme\_light()



# avg\_ts\_plot  
ggsave("Outputs/task1/plot\_01\_avg\_gdp\_ts.pdf")  
  
# NOTES for improvement:   
# there are better ways of doing the annotation one is given here, using ggpmisc  
# https://stackoverflow.com/questions/51697870/how-to-annotate-line-plot-with-arrow-and-maximum-value  
# Another solution is available using ggtext package

1. Using the Africa\_country\_average dataset, create side-by-side bar graphs of constant and current GDP per capita for all countries in the Middle East and North Africa Region. Format your graphs so that they are publication quality and save them as .pdf files.

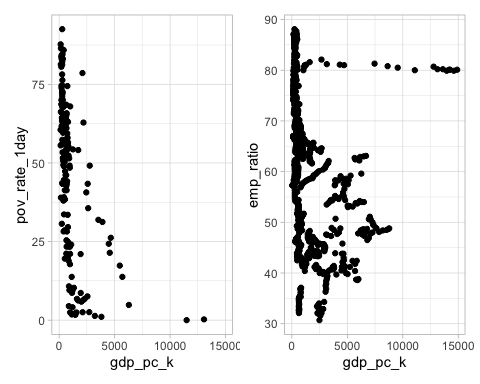
Africa\_country\_average %>%   
 dplyr::filter(Region == "Middle East & North Africa") %>%   
 dplyr::select(-Agri\_Const, -Region ) %>%   
 tidyr::pivot\_longer(cols = c(gdp\_pc\_k, gdp\_pc\_c), names\_to = "gdp\_type", values\_to = "GDP") %>%   
 dplyr::mutate(gdp\_type = dplyr::if\_else(gdp\_type == "gdp\_pc\_c", "GDP Per Capita Current", "GDP Per Capita Constant")) %>%   
 dplyr::arrange(GDP) %>%   
 ggplot(aes(x = CountryName, y = GDP)) +   
 geom\_col() +   
 facet\_grid(~gdp\_type) +   
 coord\_flip() +   
 theme\_light()



# NOTES for improvement   
# Ideally I would also arrange the plots by descending order of GDP in each plot.  
ggsave("Outputs/task1/plot\_02\_gdp\_country.pdf")

1. Generate scatter plots for: i) GDP per capita (constant) on Poverty rate(1day) ii) GDP per capita(constant) on Employment ratio. Interpret these graphs and format them as a single pdf file.

gdp\_k\_vs\_pov\_rate <- Africa\_gdp %>%   
 ggplot(aes(x = gdp\_pc\_k, y = pov\_rate\_1day)) +   
 geom\_point() +   
 theme\_light()  
  
gdp\_k\_vs\_emp\_ratio <- Africa\_gdp %>%   
 ggplot(aes(x = gdp\_pc\_k, y = emp\_ratio)) +   
 geom\_point() +   
 theme\_light()   
  
gdp\_k\_vs\_pov\_rate + gdp\_k\_vs\_emp\_ratio



ggsave("Outputs/task1/plot\_03\_gdp\_scatter.pdf")

### 1.3 Causal Inference

1. Using the main dataset (Africa\_merged), run the following linear regression and create a single table in Excel with the coefficients and main statistics (R2, no. of obs). Name the Excel file “Poverty regression”.
2. Poverty rate (1 day) on GDP (constant) per capita. Interpret the result. Would you say that the estimated effect is causal in nature? Explain. For the above regression, write out a simple econometric model.

model <- lm(gdp\_pc\_k ~ pov\_rate\_1day, data = Africa\_merged)  
  
# regular model parameters  
parameters::model\_parameters(model) %>%   
 gt::gt()

| Parameter | Coefficient | SE | CI | CI\_low | CI\_high | t | df\_error | p |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (Intercept) | 2612.48379 | 230.73122 | 0.95 | 2156.95784 | 3068.00973 | 11.322628 | 167 | 2.040015e-22 |
| pov\_rate\_1day | -33.16657 | 4.43376 | 0.95 | -41.92002 | -24.41313 | -7.480463 | 167 | 4.009251e-12 |

# saving model as .csv  
parameters::model\_parameters(model) %>%   
 gt::gt() %>%   
 write.csv("Outputs/task1/Poverty regression.csv")  
  
# library(see)  
# performance::check\_model(model). # supposed to print diagnostics graphs but isn't working at the moment  
  
# report::report\_text(model)  
report::report(model)

## We fitted a linear model (estimated using OLS) to predict gdp\_pc\_k with  
## pov\_rate\_1day (formula: gdp\_pc\_k ~ pov\_rate\_1day). The model explains a  
## statistically significant and moderate proportion of variance (R2 = 0.25, F(1,  
## 167) = 55.96, p < .001, adj. R2 = 0.25). The model's intercept, corresponding  
## to pov\_rate\_1day = 0, is at 2612.48 (95% CI [2156.96, 3068.01], t(167) = 11.32,  
## p < .001). Within this model:  
##   
## - The effect of pov rate 1day is statistically significant and negative (beta =  
## -33.17, 95% CI [-41.92, -24.41], t(167) = -7.48, p < .001; Std. beta = -0.50,  
## 95% CI [-0.63, -0.37])  
##   
## Standardized parameters were obtained by fitting the model on a standardized  
## version of the dataset. 95% Confidence Intervals (CIs) and p-values were  
## computed using a Wald t-distribution approximation.

Simple econometric model:

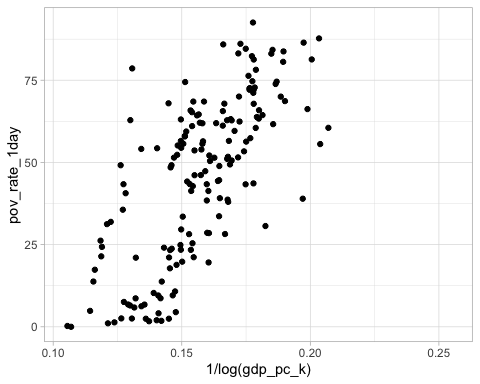
1. Suppose you estimate the model using OLS and obtain a coefficient of 0.1 with a standard error of 0.02. Interpret this result.

The coefficient measures the effect the predictor has on the target variable. The coefficient or parameter estimate depicts the slope term in the model is saying that for every 1 unit increase in the predictor, the target variable goes up by 0.1 units.

The coefficient Standard Error measures the average amount that the coefficient estimates vary from the actual average value of our response variable. Ideally we want a lower SE relative to its coefficients. In this example, for every 1 unit increase in the predictor, the value of target variable can vary upto .02 units. The Standard Error can be used to compute an estimate of the expected difference in case we ran the model again and again. In other words, we can say that the required distance for a car to stop can vary by 0.4155128 feet. The Standard Errors can also be used to compute confidence intervals.

1. The effect of GDP per capita on Poverty is likely non-linear. Explain how you would modify the above model to estimate any potential non-linearities.

Africa\_merged %>%   
 ggplot(aes(x = 1/log(gdp\_pc\_k), y = pov\_rate\_1day)) +   
 geom\_point() +   
 theme\_light()



Reciprocal of log transformation makes the relation nearly linear in this case.

1. Create a categorical variable with the levels of IncomeGroup, where 1 is poorest and 4 is wealthiest (call it IncGrp). Run the following regressions. Create a single table in Excel with each column showing the coefficients for regressions (a) and (b) and main statistics (R2, no. of obs). Name the Excel file “Income group regressions 1 and 2.xlsx (csv)”. Create another table with the marginal effects of regression (c). Name the Excel file “Income group regression 3”. The models to be run are:
2. OLS of Employment ratio on dummies for each Income group
3. OLS of Employment ratio on dummies for each Income group, and an interaction term between income groups and constant GDP per capita.
4. Create a dummy variable identifying the rich countries (those with values of IncGrp = 3 and IncGrp = 4) and zero for the rest (name this variable rich). Run a probit of rich on Employment ratio and total workers (wrks) [you can ignore the time dimension of the dataset so just pool all observations across years].

Africa\_merged\_v2 <- Africa\_merged %>%   
 dplyr::mutate(IncGrp = dplyr::case\_when(  
 IncomeGroup == "Low income" ~ 1,   
 IncomeGroup == "Lower middle income" ~ 2,  
 IncomeGroup == "Upper middle income" ~ 3,   
 IncomeGroup == "High income: nonOECD" ~ 4,  
 TRUE ~ 0),  
 rich = dplyr::if\_else(IncomeGroup %in% c(3,4),1, 0))  
  
  
# Africa\_merged %>%   
# modelbased::describe\_nonlinear(gdp\_pc\_c)  
  
# (a)  
ols\_model <- lm(data = Africa\_merged\_v2, formula = emp\_ratio ~ IncGrp)  
summary(ols\_model)

##   
## Call:  
## lm(formula = emp\_ratio ~ IncGrp, data = Africa\_merged\_v2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -27.428 -10.056 1.153 8.425 37.497   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 74.4532 0.7516 99.05 <2e-16 \*\*\*  
## IncGrp -7.4625 0.4051 -18.42 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 12.03 on 1142 degrees of freedom  
## (1880 observations deleted due to missingness)  
## Multiple R-squared: 0.2291, Adjusted R-squared: 0.2284   
## F-statistic: 339.3 on 1 and 1142 DF, p-value: < 2.2e-16

# (b)  
ols\_interaction\_model <- lm(data = Africa\_merged\_v2, formula = emp\_ratio ~ IncGrp\*gdp\_pc\_k)  
summary(ols\_interaction\_model)

##   
## Call:  
## lm(formula = emp\_ratio ~ IncGrp \* gdp\_pc\_k, data = Africa\_merged\_v2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.255 -7.266 2.642 7.367 30.339   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 82.0920976 0.8055863 101.90 <2e-16 \*\*\*  
## IncGrp -8.7870782 0.5266457 -16.68 <2e-16 \*\*\*  
## gdp\_pc\_k -0.0161919 0.0009272 -17.46 <2e-16 \*\*\*  
## IncGrp:gdp\_pc\_k 0.0048457 0.0002587 18.73 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.44 on 1106 degrees of freedom  
## (1914 observations deleted due to missingness)  
## Multiple R-squared: 0.4225, Adjusted R-squared: 0.421   
## F-statistic: 269.8 on 3 and 1106 DF, p-value: < 2.2e-16

# (c)  
probit\_model <- glm(formula = rich ~ emp\_ratio + wkrs, family = binomial(link = "probit"), data = Africa\_merged\_v2)  
summary(probit\_model)

##   
## Call:  
## glm(formula = rich ~ emp\_ratio + wkrs, family = binomial(link = "probit"),   
## data = Africa\_merged\_v2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.651e-06 -1.651e-06 -1.651e-06 -1.651e-06 -1.651e-06   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -6.991e+00 1.019e+04 -0.001 0.999  
## emp\_ratio 3.776e-16 1.600e+02 0.000 1.000  
## wkrs -6.066e-23 2.533e-05 0.000 1.000  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 0.0000e+00 on 1143 degrees of freedom  
## Residual deviance: 3.1173e-09 on 1141 degrees of freedom  
## (1880 observations deleted due to missingness)  
## AIC: 6  
##   
## Number of Fisher Scoring iterations: 25

# saving model as .csv  
parameters::model\_parameters(ols\_model) %>%   
 write.csv("Outputs/task1/Income group regressions 1.csv")  
  
parameters::model\_parameters(ols\_interaction\_model) %>%   
 write.csv("Outputs/task1/Income group regressions 2.csv")  
  
parameters::model\_parameters(probit\_model) %>%   
 write.csv("Outputs/task1/Income group regressions 3.csv")

1. Run a fixed-effects regression of constant GDP per capita on employment ratio with country- specific fixed effects, and create a single table in Excel. Name the Excel file “Fixed effect regression.xlsx (.csv)”.

# Run fixed effects model for the gdp panel data  
fixed\_effects\_reg\_model <- plm(gdp\_pc\_k ~ emp\_ratio, data = Africa\_merged,   
 index = c("CountryName", "Year"),   
 model = "within") #fixed model  
  
summary(fixed\_effects\_reg\_model)

## Oneway (individual) effect Within Model  
##   
## Call:  
## plm(formula = gdp\_pc\_k ~ emp\_ratio, data = Africa\_merged, model = "within",   
## index = c("CountryName", "Year"))  
##   
## Unbalanced Panel: n = 51, T = 11-22, N = 1110  
##   
## Residuals:  
## Min. 1st Qu. Median 3rd Qu. Max.   
## -6698.2830 -66.0758 -8.7988 52.4178 7160.5021   
##   
## Coefficients:  
## Estimate Std. Error t-value Pr(>|t|)  
## emp\_ratio 5.5807 14.0107 0.3983 0.6905  
##   
## Total Sum of Squares: 754320000  
## Residual Sum of Squares: 754200000  
## R-Squared: 0.00014994  
## Adj. R-Squared: -0.048047  
## F-statistic: 0.158659 on 1 and 1058 DF, p-value: 0.69047

fixed\_effects\_df <- fixef(fixed\_effects\_reg\_model)  
write.csv(fixed\_effects\_df, "Outputs/task1/Fixed effects regression.csv")