world\_bank\_project

Nick McCulloch, Cody Meagher, Stefano Musetti

2023-06-23

Table of Contents

# introduction and set up

**Research Question** What is the relationship between education and a country’s economy (gdp)

**hypothesis** Education has a positive correlation with GDP

## packages

library(pacman)

## Warning: package 'pacman' was built under R version 4.2.3

pacman::p\_load(readr, dplyr, tidyverse, data.table, knitr, lmtest, lubridate, ggplot2, gridExtra, shiny, sf, ggmap, maps)  
  
#install.packages("mapdata")  
  
library(mapdata)

## Warning: package 'mapdata' was built under R version 4.2.3

# packages considered but not used  
#fpp2, zoo, pscl

## data source

Data was provided by the world bank, World Development Indicators-DataBank. Specific fields of interest were selected and pulled for all countries and regions for years 1960-2022.

[World Bank Site](https://databank.worldbank.org/source/world-development-indicators)

## loading data

#wb1 <- read\_csv("wb.csv", na = "NA")  
  
wb1 <- fread("wb.csv", header = TRUE, na.strings = '"NA"')  
  
#wb\_nums <- wb1[,3:65]  
  
#wb2<- unique(wb\_nums$`1960`)  
  
sapply(wb1, class)

## Country Name Series Name 1960 1961 1962 1963   
## "character" "character" "numeric" "numeric" "numeric" "numeric"   
## 1964 1965 1966 1967 1968 1969   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 1970 1971 1972 1973 1974 1975   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 1976 1977 1978 1979 1980 1981   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 1982 1983 1984 1985 1986 1987   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 1988 1989 1990 1991 1992 1993   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 1994 1995 1996 1997 1998 1999   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 2000 2001 2002 2003 2004 2005   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 2006 2007 2008 2009 2010 2011   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 2012 2013 2014 2015 2016 2017   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## 2018 2019 2020 2021 2022   
## "numeric" "numeric" "numeric" "numeric" "logical"

summary\_wb1 <- summary(wb1)  
  
wdi\_econ\_only <- fread("WDI\_econ\_only.csv", header = TRUE)  
  
#used later on  
cols4swap <- read\_csv("wb\_cols4swap.csv")  
  
# used later on  
new\_countries <- read\_csv("country\_list\_no\_regions.csv")

# pre-processing and cleaning

## transforming data - pivots

#str(wb)  
  
colnames\_wb1 <- colnames(wb1)  
  
colnames\_wb1 <- colnames\_wb1[3:65]  
  
wb2 <- wb1 %>%  
 pivot\_longer(cols = all\_of(colnames\_wb1), names\_to = "year", values\_to = "stats")  
  
str(wb2)

## tibble [637,119 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ Country Name: chr [1:637119] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...  
## $ Series Name : chr [1:637119] "Literacy rate, adult female (% of females ages 15 and above)" "Literacy rate, adult female (% of females ages 15 and above)" "Literacy rate, adult female (% of females ages 15 and above)" "Literacy rate, adult female (% of females ages 15 and above)" ...  
## $ year : chr [1:637119] "1960" "1961" "1962" "1963" ...  
## $ stats : num [1:637119] NA NA NA NA NA NA NA NA NA NA ...

wb3 <- wb2  
  
wb3$year <- as.numeric(wb3$year)  
  
wb <- wb3  
  
rm("wb1","wb2","wb3")  
  
wbdt <- data.table(wb)  
  
# sanity check  
all.equal(wbdt,wb, check.attributes = FALSE)

## [1] TRUE

# changing colnames  
colnames(wbdt) <- c("country","series","year","stats")  
  
#sanity check  
sanity\_check <- wb[wb$`Series Name` == "GDP (constant 2015 US$)" & wb$`Country Name` == "Somalia",]  
rm(sanity\_check)  
rm(wb)

## extracting info

series\_list <- unique(wbdt$series)  
  
years <- unique(wbdt$year)

## reducing string size for series

length(series\_list)

## [1] 39

#cols4swap <- read\_csv("wb\_cols4swap.csv")  
  
cols4swap$og\_cols[38]

## [1] "GNI (constant 2015 US$)"

series\_list[39]

## [1] ""

for(i in 1:length(cols4swap$og\_cols)){  
 wbdt[series == cols4swap$og\_cols[i],series := cols4swap$new\_cols[i]]  
}

## checking NA’s

summary(wbdt$series[wbdt$series == "literacy\_af"])

## Length Class Mode   
## 16758 character character

paste("total # NAs literacy\_af:",sum(is.na(wbdt[wbdt$series == "literacy\_af",])))

## [1] "total # NAs literacy\_af: 13972"

summary(wbdt$series[wbdt$series == "gdp\_constant"])

## Length Class Mode   
## 16758 character character

paste("total # NAs gdp\_constant:",sum(is.na(wbdt[wbdt$series == "gdp\_constant",])))

## [1] "total # NAs gdp\_constant: 4232"

#wbdt[series == "Literacy rate, adult female (% of females ages 15 and above)",series := "literacy\_AF"]

## pivoting table

names\_list <- cols4swap$new\_cols  
  
#wbdtb<-wbdt  
  
#wbdt<-wbdtb  
  
# data check  
temp <- {wbdt} %>%  
 group\_by(country, year, series) %>%  
 summarise(n = n(), .groups = "drop")  
  
rm(temp)  
  
# dropping blank rows  
wbdt <- wbdt %>%  
 filter(year != "" | country != "")  
  
wbdt <- wbdt %>%  
 filter(series != "")  
  
# dropping blank rows  
#wbdt <- wbdt[!is.null(wbdt$series),]  
  
nadt <- wbdt %>% pivot\_wider(names\_from = series, values\_from = stats)

## counting NA’s by column

nas <- summary(nadt)  
  
nas <- data.frame(sapply(nadt, function(x) sum(is.na(x))))  
  
nas$cols <- row.names(nas)  
  
colnames(nas) <- c("NA\_Count","Cols")  
  
rownames(nas) <- NULL  
  
head(nas)

## NA\_Count Cols  
## 1 0 country  
## 2 0 year  
## 3 13972 literacy\_af  
## 4 13969 literacy\_am  
## 5 13967 literacy\_at  
## 6 13931 literacy\_ygpi

tail(nas)

## NA\_Count Cols  
## 35 4297 gdp\_growth  
## 36 4232 gdppc\_constant  
## 37 4297 gdppc\_growth  
## 38 9732 gnipc\_constant  
## 39 9606 gni\_growth  
## 40 9710 gni\_constant

## dropping rows without key variables

*source:* [*https://bookdown.org/rwnahhas/IntroToR/convert-numeric-to-binary.html*](https://bookdown.org/rwnahhas/IntroToR/convert-numeric-to-binary.html)

#rm(gdp\_only, nacat)  
  
gdp\_only <- nadt[,c("country", "year", "gdp\_constant")]  
  
gdp\_only$nacat <- as.numeric(is.na(gdp\_only$gdp\_constant))  
  
gdp\_filtered <- gdp\_only[gdp\_only$nacat == 0,]  
  
year\_filt <- data.frame(table(gdp\_filtered$year))  
  
head(year\_filt)

## Var1 Freq  
## 1 1960 118  
## 2 1961 123  
## 3 1962 123  
## 4 1963 123  
## 5 1964 123  
## 6 1965 130

paste("max observations: ", max(year\_filt$Freq))

## [1] "max observations: 258"

paste("min observations: ", min(year\_filt$Freq))

## [1] "min observations: 118"

country\_filt <- data.frame(table(gdp\_filtered$country))  
  
head(country\_filt$Freq)

## [1] 20 62 62 42 62 20

paste("max observations: ", max(country\_filt$Freq))

## [1] "max observations: 62"

paste("min observations: ", min(country\_filt$Freq))

## [1] "min observations: 1"

all\_filt <- data.frame(table(gdp\_filtered$country,gdp\_filtered$year))  
  
wbdt\_wide <- nadt

*original note* The data that came back from the above was weird. It indicated NA’s in recent years for big countries so testing again with a similar data set.

*explanation* It turns out a mistake earlier in the code led to a mistake loading the error, which has been corrected. This piece of code was included to highlight the processed and methods used by the team to screen for issues.

wdi\_econ\_only$nacat <- as.numeric(is.na(wdi\_econ\_only$`GDP (constant 2015 US$) [NY.GDP.MKTP.KD]`))  
  
wdi\_econ\_only <- wdi\_econ\_only[,c(1,2,4)]  
  
wdi\_gdp\_filtered <- wdi\_econ\_only[wdi\_econ\_only$nacat == 0,]  
  
wdi\_by\_year <- data.frame(table(wdi\_gdp\_filtered$Time))

## spot check - revealing unwanted data points

#finding highest gdp of all time (adjusted for inflation)  
max(wbdt\_wide$gdp\_constant, na.rm = TRUE)

## [1] 86860283231171

check\_var <- max(wbdt\_wide$gdp\_constant, na.rm = TRUE)  
  
#extracting row with highest gdp  
temp<- data.table(wbdt\_wide)  
  
temp[gdp\_constant == check\_var]

## country year literacy\_af literacy\_am literacy\_at literacy\_ygpi literacy\_yf  
## 1: World 2021 NA NA NA NA NA  
## literacy\_ym literacy\_yt edat\_ba\_f edat\_ba\_m edat\_ba\_t edat\_ls\_f edat\_ls\_m  
## 1: NA NA NA NA NA NA NA  
## edat\_ls\_t edat\_ps\_f edat\_ps\_m edat\_ps\_t edat\_prim\_f edat\_prim\_m edat\_prim\_t  
## 1: NA NA NA NA NA NA NA  
## edat\_tert\_f edat\_tert\_m edat\_tert\_t edat\_us\_f edat\_us\_m edat\_us\_t edat\_ma\_f  
## 1: NA NA NA NA NA NA NA  
## edat\_ma\_m edat\_ma\_t edat\_doc\_f edat\_doc\_m edat\_doc\_t gdp\_constant  
## 1: NA NA NA NA NA 86860283231171  
## gdp\_growth gdppc\_constant gdppc\_growth gnipc\_constant gni\_growth  
## 1: 5.874 11011 4.969 11041 6.207  
## gni\_constant  
## 1: 87098456463007

rm(temp)

The spot check revealed that global and regional aggregates had been included in the data set. The combined GDP of the earth is quite the outlier. So the next section removes these rows.

#creating list of current vars in country field  
cur\_countries <- unique(wbdt\_wide$country)  
length(cur\_countries) #266

## [1] 266

#creating list of new countries from new data set. Read in at the top  
##new\_countries <- read\_csv("country\_list\_no\_regions.csv")  
  
length(new\_countries$Country\_Name) #217

## [1] 217

new\_countries <- new\_countries[,-1]  
  
#is.data.table(wbdt)  
  
wbdt <- wbdt[country %in% c(new\_countries$Country\_Name),]  
# sanity check  
#length(unique(wbdt$country))  
#length(unique(wbdt\_wide$country))  
  
wbdt\_wide <- data.table(wbdt\_wide)  
  
wbdt\_wide <- wbdt\_wide[country %in% c(new\_countries$Country\_Name),]

## transforming countries to factors

wbdt$country <- as.factor(wbdt$country)  
class(wbdt$country)

## [1] "factor"

wbdt\_wide$country <- as.factor(wbdt\_wide$country)

## dropping rows without key variables

Dropping rows with NA’s in the key variables, which in this case are, gni\_constant and gnipc\_constant.

#first getting a new NA count  
  
na\_by\_col <- wbdt\_wide %>% summarise(across(everything(), ~ sum(is.na(.))))  
  
# and the inverse  
vals\_by\_col <- wbdt\_wide %>% summarise(across(everything(), ~ sum(!is.na(.))))  
  
paste(colnames(vals\_by\_col), ":", vals\_by\_col)

## [1] "country : 13671" "year : 13671" "literacy\_af : 1067"   
## [4] "literacy\_am : 1067" "literacy\_at : 1070" "literacy\_ygpi : 1108"   
## [7] "literacy\_yf : 1185" "literacy\_ym : 1108" "literacy\_yt : 1111"   
## [10] "edat\_ba\_f : 520" "edat\_ba\_m : 520" "edat\_ba\_t : 523"   
## [13] "edat\_ls\_f : 1223" "edat\_ls\_m : 1223" "edat\_ls\_t : 1240"   
## [16] "edat\_ps\_f : 866" "edat\_ps\_m : 866" "edat\_ps\_t : 878"   
## [19] "edat\_prim\_f : 992" "edat\_prim\_m : 992" "edat\_prim\_t : 998"   
## [22] "edat\_tert\_f : 1084" "edat\_tert\_m : 1084" "edat\_tert\_t : 1093"   
## [25] "edat\_us\_f : 1168" "edat\_us\_m : 1168" "edat\_us\_t : 1176"   
## [28] "edat\_ma\_f : 402" "edat\_ma\_m : 402" "edat\_ma\_t : 404"   
## [31] "edat\_doc\_f : 324" "edat\_doc\_m : 324" "edat\_doc\_t : 325"   
## [34] "gdp\_constant : 9857" "gdp\_growth : 9840" "gdppc\_constant : 9857"  
## [37] "gdppc\_growth : 9840" "gnipc\_constant : 5458" "gni\_growth : 5632"   
## [40] "gni\_constant : 5480"

rm(vals\_by\_col, na\_by\_col)

# now dropping NAs  
wide\_narm <- wbdt\_wide[!is.na(gnipc\_constant),]  
  
wb\_narm <- wbdt[!is.na(stats),]

*str commented out because of space constraints*

#str(wbdt\_wide)  
#str(wb\_narm)

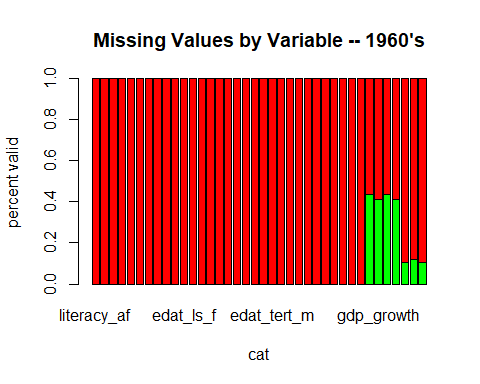
## creating data and plot to be used in shiny

Creating df to be used later in shiny

objs <- ls()  
  
if("temp" %in% objs){rm(temp)}  
if("data" %in% objs){rm(data)}  
rm(objs)  
  
temp <- wbdt\_wide[,!c("country")]  
  
data <- data.table(temp)  
  
# Function to calculate decade  
get\_decade <- function(year) {  
 floor(year / 10) \* 10  
}  
  
# Add decade column to the data table  
data[, decade := get\_decade(year)]  
  
data <- data[, !"year"]  
  
temp <- data[, lapply(.SD, function(x) as.integer(!is.na(x) & !is.nan(x))), .SDcols = -"decade"]  
  
data <- cbind(data$decade, temp)  
  
colnames(data)[1] <- "decade"  
  
#data\_aggregated <- data[, lapply(.SD, sum), by = decade]  
  
#  
data\_sum <- data[, lapply(.SD, sum), by = decade]  
  
temp <- data[, lapply(.SD, function(x) factor(x)), .SDcols = -"decade"]  
  
data <- cbind(data$decade, temp)  
  
colnames(data)[1] <- "decade"  
  
data\_total <- data[, lapply(.SD, length), by = decade]  
  
data\_perc <- data\_sum/data\_total  
data\_perc$decade <- data\_sum$decade  
  
rm(data, temp)

*A plot using the data above that is used as the basis for shiny GIF*

temp <- data\_perc[,-1]  
  
temp2<- as.numeric(temp[1,])  
  
Values <- matrix(c(temp2,1-temp2), nrow = 2, ncol = 38, byrow = TRUE)  
  
colnames\_perc <- colnames(data\_perc)  
colnames\_perc <- colnames\_perc[-1]  
  
colnames(Values) <- colnames\_perc  
  
colors = c("green","red")  
  
barplot(Values, main = "Missing Values by Variable -- 1960's",   
 xlab = "cat", ylab = "percent valid", col = colors, names.arg = colnames(Values))



par(mar = c(8, 4.1, 4.1, 2.1), las=2)  
  
rm(temp,temp2,Values,colnames\_perc)

## dropping unneeded df’s

# dropping wdi\_econ\_only, as its no longer needed  
rm(wdi\_econ\_only, wdi\_gdp\_filtered, wdi\_by\_year, year\_filt, nas, nadt, country\_filt, all\_filt, cols4swap, gdp\_only, gdp\_filtered, new\_countries, i, cur\_countries, check\_var, colnames\_wb1, years, series\_list, names\_list, summary\_wb1)

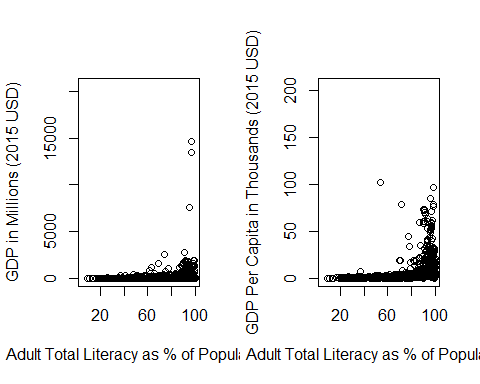
The remaining columns are wbdt: a long format data set, wbdt\_wide: the wide version of wbdt, wb\_narm: wbdt where all rows with NA’s have been removed (less impactful in this case because each field has its own row), and wide\_narm: where only rows with NA’s in gnipc\_constant have been removed.

# Exploratory Analysis

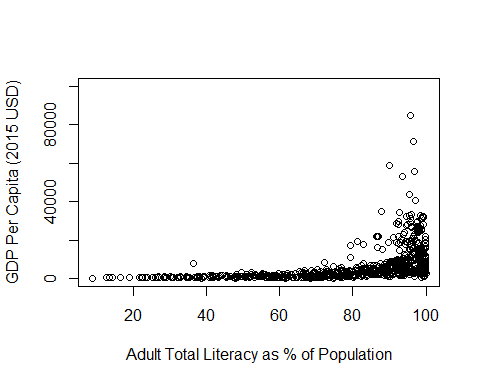
Here we perform initial analyses and visualizations to get a sense of the data and spot potential issues.

## quick test plots

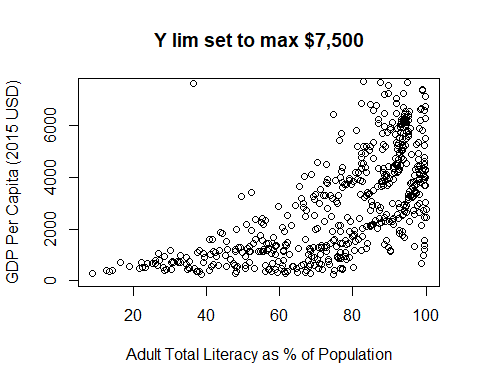
par(mfrow = c(1,2))  
  
plot(wbdt\_wide$literacy\_at, wbdt\_wide$gdp\_constant/1000000000, ylab = "GDP in Millions (2015 USD)", xlab = "Adult Total Literacy as % of Population")  
  
plot(x = wbdt\_wide$literacy\_at, y = wbdt\_wide$gdppc\_constant/1000, ylab = "GDP Per Capita in Thousands (2015 USD)", xlab = "Adult Total Literacy as % of Population")

 The plot seems to have significant outliers making it difficult to read.

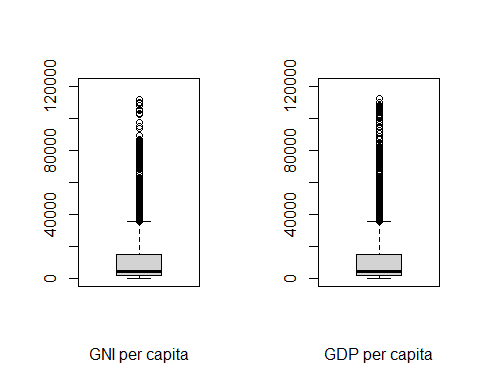
# recreating plot, limiting money range to 100/100K min/max  
plot(wide\_narm$literacy\_at, wide\_narm$gdppc\_constant, ylab = "GDP Per Capita (2015 USD)", xlab = "Adult Total Literacy as % of Population", ylim = c(100,100000))

 Although somewhat improved, the outliers are still a problem.

# recreating plot, limiting money range to 100/75K min/max  
plot(wide\_narm$literacy\_at, wide\_narm$gdppc\_constant, ylab = "GDP Per Capita (2015 USD)", xlab = "Adult Total Literacy as % of Population", ylim = c(100,7500), main = "Y lim set to max $7,500")



par(mfrow = c(1,2))  
boxplot(wide\_narm$gnipc\_constant, ylim = c(0,120000), xlab = "GNI per capita")  
boxplot(wide\_narm$gdppc\_constant, ylim = c(0,120000), xlab = "GDP per capita")

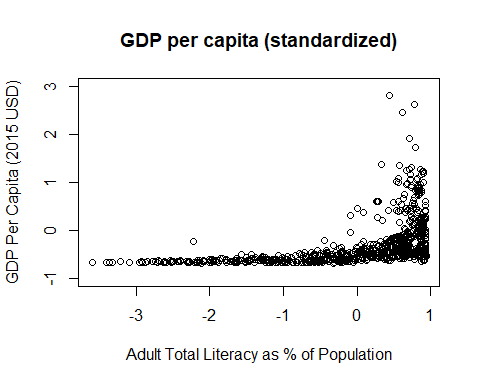
 box plots of GDP and GNI indicate the same pattern even with limits set on y.

### standardizing data

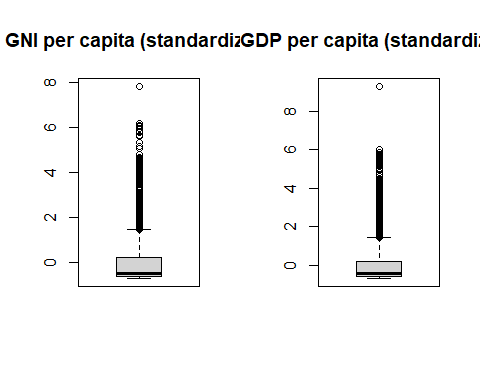
Standardizing the data highlights just how far from the norm the outliers are.

z\_wide <- as.data.frame(scale(wide\_narm[,!c("country","year")]))  
  
z\_wide <- cbind(wide\_narm$country,wide\_narm$year,z\_wide)  
  
colnames(z\_wide)[1] <- "country"  
colnames(z\_wide)[2] <- "year"

plot(z\_wide$literacy\_at, z\_wide$gdppc\_constant, ylab = "GDP Per Capita (2015 USD)", xlab = "Adult Total Literacy as % of Population", main = "GDP per capita (standardized)", ylim = c(-1,3))



par(mfrow = c(1,2))  
boxplot(z\_wide$gnipc\_constant, main = "GNI per capita (standardized)")  
boxplot(z\_wide$gdppc\_constant, main = "GDP per capita (standardized)")



### regression

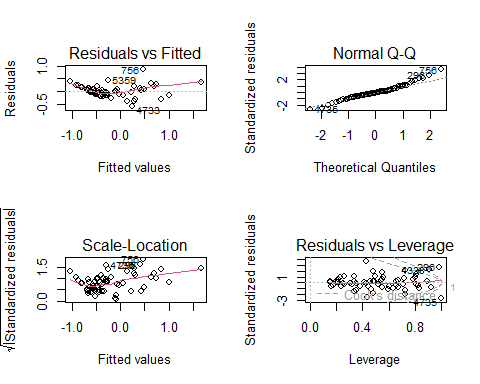
prelim\_df <- subset(z\_wide, select = -c(gdp\_constant, gdp\_growth, gdppc\_growth, gni\_growth,gni\_constant, country))  
  
prelim\_df1 <- subset(prelim\_df, select = -c(gdppc\_constant, year))  
  
prelim\_df2 <- subset(prelim\_df, select = -gnipc\_constant)  
  
lm\_prelim1 <- lm(gnipc\_constant ~., data = prelim\_df1)  
  
lm\_prelim2 <- lm(gdppc\_constant ~., data = prelim\_df2)  
  
summary(lm\_prelim1)

##   
## Call:  
## lm(formula = gnipc\_constant ~ ., data = prelim\_df1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.5802 -0.1348 -0.0190 0.0948 0.8497   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.211 0.393 0.54 0.5955   
## literacy\_af -9.841 7.489 -1.31 0.1985   
## literacy\_am -7.358 4.987 -1.48 0.1502   
## literacy\_at 17.172 12.273 1.40 0.1717   
## literacy\_ygpi 1.049 0.872 1.20 0.2379   
## literacy\_yf 12.630 9.299 1.36 0.1842   
## literacy\_ym 11.524 7.563 1.52 0.1377   
## literacy\_yt -24.553 16.614 -1.48 0.1495   
## edat\_ba\_f 9.390 13.472 0.70 0.4910   
## edat\_ba\_m 6.074 10.633 0.57 0.5719   
## edat\_ba\_t -15.077 23.619 -0.64 0.5279   
## edat\_ls\_f -17.155 20.984 -0.82 0.4198   
## edat\_ls\_m -13.512 18.518 -0.73 0.4711   
## edat\_ls\_t 30.343 39.455 0.77 0.4477   
## edat\_ps\_f -26.791 8.065 -3.32 0.0023 \*\*  
## edat\_ps\_m -18.482 6.722 -2.75 0.0099 \*\*  
## edat\_ps\_t 44.303 14.602 3.03 0.0048 \*\*  
## edat\_prim\_f 5.187 8.425 0.62 0.5426   
## edat\_prim\_m 3.853 7.200 0.54 0.5964   
## edat\_prim\_t -8.849 15.578 -0.57 0.5741   
## edat\_tert\_f -3.733 11.493 -0.32 0.7475   
## edat\_tert\_m -1.692 9.352 -0.18 0.8576   
## edat\_tert\_t 5.299 20.233 0.26 0.7951   
## edat\_us\_f 49.563 25.171 1.97 0.0579 .   
## edat\_us\_m 40.668 23.353 1.74 0.0915 .   
## edat\_us\_t -89.519 48.297 -1.85 0.0734 .   
## edat\_ma\_f 14.300 13.539 1.06 0.2991   
## edat\_ma\_m 10.908 11.539 0.95 0.3518   
## edat\_ma\_t -24.958 24.776 -1.01 0.3216   
## edat\_doc\_f -0.976 4.625 -0.21 0.8342   
## edat\_doc\_m -1.755 6.348 -0.28 0.7841   
## edat\_doc\_t 3.434 10.865 0.32 0.7541   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.315 on 31 degrees of freedom  
## (5395 observations deleted due to missingness)  
## Multiple R-squared: 0.848, Adjusted R-squared: 0.697   
## F-statistic: 5.59 on 31 and 31 DF, p-value: 0.00000336

#summary(lm\_prelim2)

### residuals plots and BP test

par(mfrow = c(2,2))  
  
plot(lm\_prelim1)



#plot(lm\_prelim2) #results similar to lm\_prelim1

The residuals/fitted plot shows large deviation from linearity to the right. The QQ plot shows light tails, indicating more data at the extremes compared to a normal QQ plot. The residuals vs leverage confirms the presence of significant outliers. The scale location plot is neither horizontal nor evenly spread, indicating heteroskedasticity, however, this wasn’t conclusively confirmed by the BP tests below.

bptest(lm\_prelim1, data = prelim\_df1)

##   
## studentized Breusch-Pagan test  
##   
## data: lm\_prelim1  
## BP = 42, df = 31, p-value = 0.09

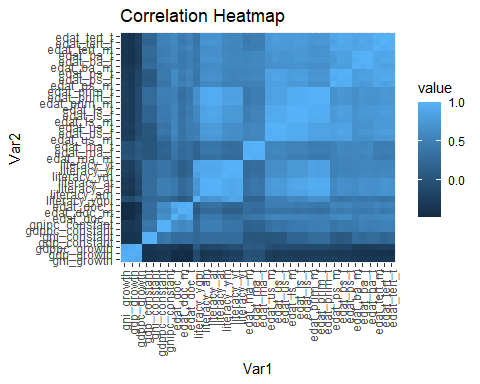
bptest(lm\_prelim2, data = prelim\_df2)

##   
## studentized Breusch-Pagan test  
##   
## data: lm\_prelim2  
## BP = 45, df = 32, p-value = 0.07

### multicollinearity checks

Given the nature of the data some of the variables are guaranteed to suffer from some multicollinearity, for example gdp\_growth and gni\_growth or literacy and primary education attainment. The correlation heat-map below explores this.

# creating correlation matrix  
  
multi <- subset(wide\_narm, select = -c(country, year))  
  
multi <- drop\_na(multi)  
  
corr\_mat <- round(cor(multi),2)  
  
# sorting matrix for easier interpretation  
dist <- as.dist((1-corr\_mat)/2)  
  
# clustering the dist matrix  
hclust <- hclust(dist)  
corr\_mat <- corr\_mat[hclust$order, hclust$order]  
  
# reduce the size of correlation matrix  
melted\_corr\_mat <- reshape2::melt(corr\_mat)  
  
#fwrite(melted\_corr\_mat, "melted\_corr\_mat.csv")  
  
#plotting the correlation heat-map  
ggplot(data = melted\_corr\_mat, aes(x = Var1, y = Var2, fill = value)) +  
 geom\_tile() + labs(title = "Correlation Heatmap")+  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))



**Correlation Analysis**

the heat-map highlights areas of correlation. This is explored further with the correlation test below.

cor\_test <- cor.test(wide\_narm$gni\_constant, wide\_narm$gdp\_constant, use = "complete.obs", method = "pearson")  
print(cor\_test)

##   
## Pearson's product-moment correlation  
##   
## data: wide\_narm$gni\_constant and wide\_narm$gdp\_constant  
## t = 5463, df = 5456, p-value <0.0000000000000002  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9999 0.9999  
## sample estimates:  
## cor   
## 0.9999

**Summary and Explanation of Results**

Cor: The correlation coefficient tells us the strength and direction of the linear relationship between the two variables. our correlation coefficient, 0.9999, indicates a near perfect correlation between GNI and GPD.

P-value: The p-value tests the likelihood the null hypothesis is true (that there is no correlation).Our p-value is way below 0.05, firmly disproving the null hypothesis, which means it’s extremely likely that the variables are in fact correlated.

t: This is the t-value, which is used to calculate the p-value that’s described above.

df: This is the number of data points used in the cor.test.

95% confidence interval: This means that if we were to run this test 20 times in 19 of them the right answer would fall in the range we’ve constructed.

In summary, our analysis indicates that there is a very strong positive correlation between GNI and GDP. This is just one example of the significant multicollinearity that we expected and which is confirmed by the cor.test and the heat-map. High correlation between predictor variables means they’re not truly independent and that without adjustments we are unable to say what portion of the data is explained by one variable vs a correlated one.

# Refined Analysis

### repeating LM test with single year

By using a single year (and ad-ho variable selection) we can explore the data with less multicollinearity and less impact by any time based trend.

prelim\_df <- subset(z\_wide, select = -c(gdp\_constant, gdp\_growth, gdppc\_growth, gni\_growth,gni\_constant, country))  
  
prelim\_df1 <- subset(prelim\_df, subset = year == 2015, select = -gdppc\_constant)  
prelim\_df2 <- subset(prelim\_df, subset = year == 2015, select = -gnipc\_constant)  
  
prelim\_df1 <- subset(prelim\_df1, select = -year)  
prelim\_df2 <- subset(prelim\_df2, select = -year)  
  
nas <- data.frame(sapply(prelim\_df1, function(x) sum(is.na(x))))  
  
  
prelim\_df1 <- data.frame(prelim\_df1)  
  
lm\_prelim1 <- lm(gdppc\_constant ~ edat\_us\_t, data = prelim\_df2)  
lm\_prelim2 <- lm(gnipc\_constant ~ edat\_us\_t + edat\_us\_f + edat\_us\_m + edat\_tert\_t, data = prelim\_df1)  
  
summary(lm\_prelim1)

##   
## Call:  
## lm(formula = gdppc\_constant ~ edat\_us\_t, data = prelim\_df2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.114 -0.939 -0.138 0.701 4.636   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.490 0.162 3.03 0.00358 \*\*   
## edat\_us\_t 0.692 0.167 4.14 0.00011 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.24 on 62 degrees of freedom  
## (136 observations deleted due to missingness)  
## Multiple R-squared: 0.216, Adjusted R-squared: 0.204   
## F-statistic: 17.1 on 1 and 62 DF, p-value: 0.000108

summary(lm\_prelim2)

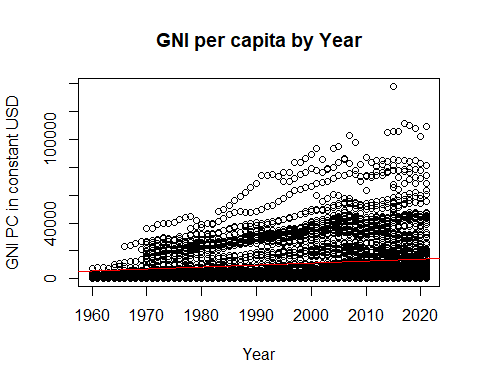
##   
## Call:  
## lm(formula = gnipc\_constant ~ edat\_us\_t + edat\_us\_f + edat\_us\_m +   
## edat\_tert\_t, data = prelim\_df1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.082 -0.485 -0.182 0.635 3.086   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.563 0.190 2.97 0.0044 \*\*  
## edat\_us\_t -17.166 16.223 -1.06 0.2947   
## edat\_us\_f 8.976 8.291 1.08 0.2838   
## edat\_us\_m 8.441 8.066 1.05 0.3000   
## edat\_tert\_t 0.629 0.248 2.54 0.0141 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.12 on 54 degrees of freedom  
## (141 observations deleted due to missingness)  
## Multiple R-squared: 0.338, Adjusted R-squared: 0.289   
## F-statistic: 6.89 on 4 and 54 DF, p-value: 0.000148

The single year tests suffer from high levels of sparsity. When only the least sparse variables are selected, some statistically significant effects can be seen (% tertiary educational attainment has a positive relationship with GNI per capita)

## time based analysis

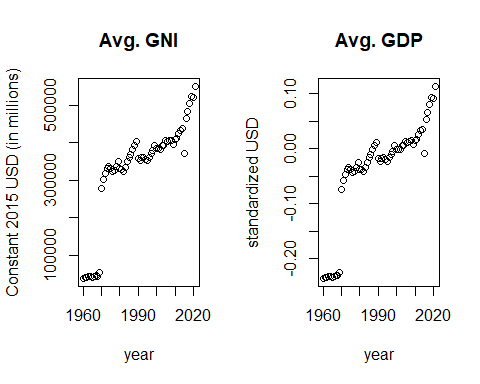
### GNI\_pc by year wtih fitted line

plot(wide\_narm$year, wide\_narm$gnipc\_constant, main="GNI per capita by Year", xlab="Year", ylab="GNI PC in constant USD")  
fit <- lm(gnipc\_constant ~ year, data = wide\_narm)  
abline(fit, col = "red")



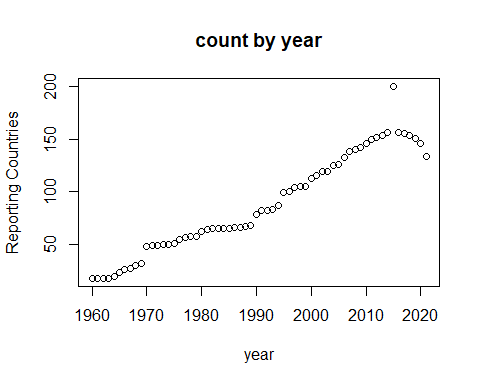
### mean GNI and GDP by year

by\_year <- wide\_narm %>%  
 group\_by(year) %>%  
 summarise(avg = mean(gni\_constant/1000000))  
  
by\_year2 <- z\_wide %>%  
 group\_by(year) %>%  
 summarise(avg = mean(gdp\_constant))  
  
par(mfrow = c(1,2))  
  
plot(by\_year, ylab = "Constant 2015 USD (in millions)", main = "Avg. GNI")  
plot(by\_year2, ylab = "standardized USD", main = "Avg. GDP")

 The plots demonstrate a clear growth in average GDP and GNI over time. However, a general growth in GDP since the 1960’s is essentially guaranteed because of the amount of population growth over the last 60 years.

It should be noted that pure totals also can’t be used because of the growth in the number of reporting countries in the world bank data set as seen below.

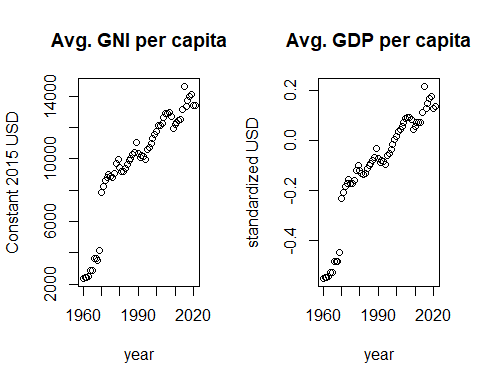
count\_byyear <- wide\_narm %>%   
 group\_by(year) %>%  
 summarise(across(everything(), ~ sum(!is.na(.))))  
  
plot(count\_byyear$year,count\_byyear$gni\_constant, ylab = "Reporting Countries", main = "count by year", xlab = "year")



### mean GNI and GDP per capita by year

Although total GNI or GDP growth is distorted by the growth in population and reporting countries, per capita means aren’t affected. Per

by\_year <- wide\_narm %>%  
 group\_by(year) %>%  
 summarise(avg = mean(gnipc\_constant))  
  
by\_year2 <- z\_wide %>%  
 group\_by(year) %>%  
 summarise(avg = mean(gdppc\_constant))  
  
par(mfrow = c(1,2))  
  
plot(by\_year, ylab = "Constant 2015 USD", main = "Avg. GNI per capita")  
plot(by\_year2, ylab = "standardized USD", main = "Avg. GDP per capita")



When grouped and averaged by year, the data shows a clear upwards trend over time even when accounting for population growth with GDP or GNI per capita.

fwrite(wide\_narm, "wide\_narm.csv")

# RShiny

### map data for shiny

The instructions at [Sharp Sight](https://www.sharpsightlabs.com/blog/map-talent-competitiveness/) were helpful in producing the map plot and data.

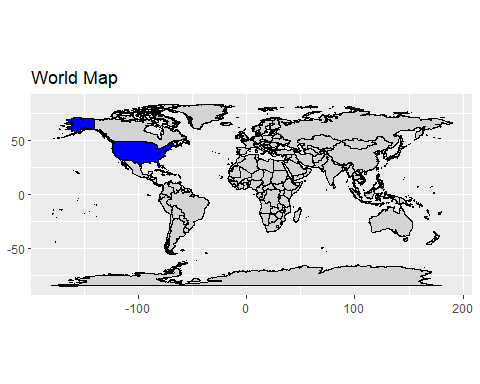
# Creating objects with country/map data  
world\_map <- map\_data('world')  
  
wb\_countries <- data.frame(country = unique(wide\_narm$country))  
  
# Checking disparities between world bank country names and world\_map names.  
anti\_join(wb\_countries, world\_map, by = c('country' = 'region'))

## country  
## 1 Antigua and Barbuda  
## 2 Bahamas, The  
## 3 Brunei Darussalam  
## 4 Cabo Verde  
## 5 Congo, Dem. Rep.  
## 6 Congo, Rep.  
## 7 Cote d'Ivoire  
## 8 Czechia  
## 9 Egypt, Arab Rep.  
## 10 Eswatini  
## 11 Gambia, The  
## 12 Hong Kong SAR, China  
## 13 Iran, Islamic Rep.  
## 14 Korea, Rep.  
## 15 Kyrgyz Republic  
## 16 Lao PDR  
## 17 Macao SAR, China  
## 18 Micronesia, Fed. Sts.  
## 19 Russian Federation  
## 20 Sint Maarten (Dutch part)  
## 21 Slovak Republic  
## 22 St. Kitts and Nevis  
## 23 St. Lucia  
## 24 St. Vincent and the Grenadines  
## 25 Syrian Arab Republic  
## 26 Trinidad and Tobago  
## 27 Turkiye  
## 28 Tuvalu  
## 29 United Kingdom  
## 30 United States  
## 31 West Bank and Gaza  
## 32 Yemen, Rep.

# printing list of country names in wold\_map  
world\_map %>%  
 group\_by(region) %>%  
 summarise() %>%  
 print(n = Inf)

## # A tibble: 252 × 1  
## region   
## <chr>   
## 1 Afghanistan   
## 2 Albania   
## 3 Algeria   
## 4 American Samoa   
## 5 Andorra   
## 6 Angola   
## 7 Anguilla   
## 8 Antarctica   
## 9 Antigua   
## 10 Argentina   
## 11 Armenia   
## 12 Aruba   
## 13 Ascension Island   
## 14 Australia   
## 15 Austria   
## 16 Azerbaijan   
## 17 Azores   
## 18 Bahamas   
## 19 Bahrain   
## 20 Bangladesh   
## 21 Barbados   
## 22 Barbuda   
## 23 Belarus   
## 24 Belgium   
## 25 Belize   
## 26 Benin   
## 27 Bermuda   
## 28 Bhutan   
## 29 Bolivia   
## 30 Bonaire   
## 31 Bosnia and Herzegovina   
## 32 Botswana   
## 33 Brazil   
## 34 Brunei   
## 35 Bulgaria   
## 36 Burkina Faso   
## 37 Burundi   
## 38 Cambodia   
## 39 Cameroon   
## 40 Canada   
## 41 Canary Islands   
## 42 Cape Verde   
## 43 Cayman Islands   
## 44 Central African Republic   
## 45 Chad   
## 46 Chagos Archipelago   
## 47 Chile   
## 48 China   
## 49 Christmas Island   
## 50 Cocos Islands   
## 51 Colombia   
## 52 Comoros   
## 53 Cook Islands   
## 54 Costa Rica   
## 55 Croatia   
## 56 Cuba   
## 57 Curacao   
## 58 Cyprus   
## 59 Czech Republic   
## 60 Democratic Republic of the Congo   
## 61 Denmark   
## 62 Djibouti   
## 63 Dominica   
## 64 Dominican Republic   
## 65 Ecuador   
## 66 Egypt   
## 67 El Salvador   
## 68 Equatorial Guinea   
## 69 Eritrea   
## 70 Estonia   
## 71 Ethiopia   
## 72 Falkland Islands   
## 73 Faroe Islands   
## 74 Fiji   
## 75 Finland   
## 76 France   
## 77 French Guiana   
## 78 French Polynesia   
## 79 French Southern and Antarctic Lands  
## 80 Gabon   
## 81 Gambia   
## 82 Georgia   
## 83 Germany   
## 84 Ghana   
## 85 Greece   
## 86 Greenland   
## 87 Grenada   
## 88 Grenadines   
## 89 Guadeloupe   
## 90 Guam   
## 91 Guatemala   
## 92 Guernsey   
## 93 Guinea   
## 94 Guinea-Bissau   
## 95 Guyana   
## 96 Haiti   
## 97 Heard Island   
## 98 Honduras   
## 99 Hungary   
## 100 Iceland   
## 101 India   
## 102 Indonesia   
## 103 Iran   
## 104 Iraq   
## 105 Ireland   
## 106 Isle of Man   
## 107 Israel   
## 108 Italy   
## 109 Ivory Coast   
## 110 Jamaica   
## 111 Japan   
## 112 Jersey   
## 113 Jordan   
## 114 Kazakhstan   
## 115 Kenya   
## 116 Kiribati   
## 117 Kosovo   
## 118 Kuwait   
## 119 Kyrgyzstan   
## 120 Laos   
## 121 Latvia   
## 122 Lebanon   
## 123 Lesotho   
## 124 Liberia   
## 125 Libya   
## 126 Liechtenstein   
## 127 Lithuania   
## 128 Luxembourg   
## 129 Madagascar   
## 130 Madeira Islands   
## 131 Malawi   
## 132 Malaysia   
## 133 Maldives   
## 134 Mali   
## 135 Malta   
## 136 Marshall Islands   
## 137 Martinique   
## 138 Mauritania   
## 139 Mauritius   
## 140 Mayotte   
## 141 Mexico   
## 142 Micronesia   
## 143 Moldova   
## 144 Monaco   
## 145 Mongolia   
## 146 Montenegro   
## 147 Montserrat   
## 148 Morocco   
## 149 Mozambique   
## 150 Myanmar   
## 151 Namibia   
## 152 Nauru   
## 153 Nepal   
## 154 Netherlands   
## 155 Nevis   
## 156 New Caledonia   
## 157 New Zealand   
## 158 Nicaragua   
## 159 Niger   
## 160 Nigeria   
## 161 Niue   
## 162 Norfolk Island   
## 163 North Korea   
## 164 North Macedonia   
## 165 Northern Mariana Islands   
## 166 Norway   
## 167 Oman   
## 168 Pakistan   
## 169 Palau   
## 170 Palestine   
## 171 Panama   
## 172 Papua New Guinea   
## 173 Paraguay   
## 174 Peru   
## 175 Philippines   
## 176 Pitcairn Islands   
## 177 Poland   
## 178 Portugal   
## 179 Puerto Rico   
## 180 Qatar   
## 181 Republic of Congo   
## 182 Reunion   
## 183 Romania   
## 184 Russia   
## 185 Rwanda   
## 186 Saba   
## 187 Saint Barthelemy   
## 188 Saint Helena   
## 189 Saint Kitts   
## 190 Saint Lucia   
## 191 Saint Martin   
## 192 Saint Pierre and Miquelon   
## 193 Saint Vincent   
## 194 Samoa   
## 195 San Marino   
## 196 Sao Tome and Principe   
## 197 Saudi Arabia   
## 198 Senegal   
## 199 Serbia   
## 200 Seychelles   
## 201 Siachen Glacier   
## 202 Sierra Leone   
## 203 Singapore   
## 204 Sint Eustatius   
## 205 Sint Maarten   
## 206 Slovakia   
## 207 Slovenia   
## 208 Solomon Islands   
## 209 Somalia   
## 210 South Africa   
## 211 South Georgia   
## 212 South Korea   
## 213 South Sandwich Islands   
## 214 South Sudan   
## 215 Spain   
## 216 Sri Lanka   
## 217 Sudan   
## 218 Suriname   
## 219 Swaziland   
## 220 Sweden   
## 221 Switzerland   
## 222 Syria   
## 223 Taiwan   
## 224 Tajikistan   
## 225 Tanzania   
## 226 Thailand   
## 227 Timor-Leste   
## 228 Tobago   
## 229 Togo   
## 230 Tonga   
## 231 Trinidad   
## 232 Tunisia   
## 233 Turkey   
## 234 Turkmenistan   
## 235 Turks and Caicos Islands   
## 236 UK   
## 237 USA   
## 238 Uganda   
## 239 Ukraine   
## 240 United Arab Emirates   
## 241 Uruguay   
## 242 Uzbekistan   
## 243 Vanuatu   
## 244 Vatican   
## 245 Venezuela   
## 246 Vietnam   
## 247 Virgin Islands   
## 248 Wallis and Futuna   
## 249 Western Sahara   
## 250 Yemen   
## 251 Zambia   
## 252 Zimbabwe

# recoding names  
wide\_narm <- wide\_narm %>% mutate(country = recode(  
 country,  
 `Antigua and Barbuda` = 'Antigua',  
 `Bahamas, The` = 'Bahamas',  
 `Brunei Darussalam` = 'Brunei',  
 `Cabo Verde` = 'Cape Verde',  
 `Congo, Dem. Rep.` = 'Democratic Republic of the Congo',  
 `Congo, Rep.` = 'Republic of Congo',  
 `Cote d'Ivoire` = 'Ivory Coast',  
 `Czechia` = 'Czech Republic',  
 `Egypt, Arab Rep.` = 'Egypt',  
 `Eswatini` = 'Swaziland',  
 `Gambia, The` = 'Gambia',  
 `Iran, Islamic Rep.` = 'Iran',  
 `Korea, Rep.` = 'South Korea',  
 `Kyrgyz Republic` = 'Kyrgyzstan',  
 `Lao PDR` = 'Lao',  
 `Micronesia, Fed. Sts.` = 'Micronesia',  
 `Russian Federation` = 'Russia',  
 `Sint Maarten (Dutch part)` = 'Saint Martin',  
 `Slovak Republic` = 'Slovakia',  
 `St. Kitts and Nevis` = 'Saint Kitts',  
 `St. Lucia` = 'Saint Lucia',  
 `St. Vincent and the Grenadines` = 'Saint Vincent',  
 `Syrian Arab Republic` = 'Syria',  
 `Trinidad and Tobago` = 'Trinidad',  
 `Turkiye` = 'Turkey',  
 `United Kingdom` = 'UK',  
 `United States` = 'USA',  
 `West Bank and Gaza` = 'Palestine',  
 `Yemen, Rep.` = 'Yemen',  
)  
)  
  
# creating test plot  
  
world\_plot <- ggplot() +  
 geom\_polygon(data = world\_map, aes(x = long, y = lat, group = group), fill = "lightgray", color = "black") +  
 geom\_polygon(data = subset(world\_map, region == "USA"), aes(x = long, y = lat, group = group), fill = "blue", color = "black") +  
 coord\_equal() +  
 labs(title = "World Map")+xlab(NULL)+ylab(NULL)  
  
world\_plot



## full RShiny code

library(pacman)  
  
pacman::p\_load(readr, dplyr, tidyverse, data.table, knitr, lmtest, lubridate, ggplot2, gridExtra, shiny, sf, ggmap, maps, cowplot)  
  
library(mapdata)  
  
wide\_narm <- fread("wide\_narm.csv")  
  
if(!is.data.table(wide\_narm)){wide\_narm <- data.table(wide\_narm)}  
  
wide\_narm$country <- as.factor(wide\_narm$country)  
  
df <- subset(wide\_narm, select = -country)  
  
world\_map <- map\_data("world")  
  
prelim\_df1 <- fread("prelim\_df1.csv")  
  
lm\_prelim1 <- lm(prelim\_df1$gnipc\_constant ~., data = prelim\_df1)  
  
melted\_corr\_mat <- fread("melted\_corr\_mat.csv")  
  
#world\_map <- left\_join( world\_map, wb\_countries, by = c('region' = 'country'))   
  
# Group data by decade  
df\_decade <- df %>%  
 mutate(decade = 10 \* floor(year / 10))  
  
ui <- fluidPage(  
   
 fluidRow(  
 titlePanel(  
 h1("World Bank Project"  
 ))  
 ),  
 fluidRow(  
 titlePanel(h3("Group: Nick McCulloch, Cody Meagher, Stefano Mesetti"  
 ))  
 ),  
   
 hr(),  
   
 fluidRow(  
 titlePanel(  
 h3("Dataset Limitations"  
 )),  
 #sidebarLayout(  
 #sidebarPanel(),  
 #mainPanel(  
 img(src="gif.gif", align = "left",height='450px',width='900px')  
 #),  
 #)  
 ),  
   
 fluidRow(  
 sidebarLayout(  
 sidebarPanel(  
 numericInput("residual\_var",  
 "Residual/Multicollinearity Plots:", min = 1, max = 6, value = 1, step = 1)),  
 mainPanel(  
 plotOutput("residual\_plot")  
 ),  
 )  
 ),  
   
 hr(),  
   
 fluidRow(  
 titlePanel(  
 h3("Interpretation"  
 )),  
 ),  
   
 fluidRow(  
 sidebarLayout(  
 sidebarPanel(  
 sliderInput(  
 "decade\_slider",  
 "Decade:",  
 min = 1960,  
 max = 2020,  
 value = 2010,  
 step = 10  
 ),  
   
 varSelectInput("scatter\_varX","select x variable",df, selected = "literacy\_at"),  
 varSelectInput("scatter\_varY","select y variable",df, selected = "gnipc\_constant"),  
 checkboxInput("smooth", "Add Regression Line", value = FALSE),  
 sliderInput(  
 "y\_lim\_slider",  
 "Max Income:",  
 min = 1000,  
 max = 200000,  
 value = 75000,  
 step = 1000  
 ),  
 ),  
 mainPanel(  
 plotOutput("scatter\_plot")  
 ))  
 ),  
   
 hr(),  
   
 fluidRow(  
 titlePanel("GNI and GDP Analysis"),  
   
 sidebarLayout(  
 sidebarPanel(  
 selectInput("country",   
 label = "Choose a country",  
 choices = unique(wide\_narm$country),  
 selected = "South Africa"  
 )  
 ),  
   
 mainPanel(  
 tabsetPanel(type = "tabs",  
 tabPanel("GNI and GDP Over Time",   
 plotOutput("timePlot"),  
 plotOutput("countryplot"))  
 )  
 )  
 )  
 ),  
 fluidRow(  
   
 ),  
 fluidRow(  
 )  
)  
  
server <- function(input, output) {  
 output$residual\_plot <- renderPlot({  
 if(input$residual\_var < 6) {  
 plot(lm\_prelim1, which = input$residual\_var)  
 } else {  
 ggplot(data = melted\_corr\_mat, aes(x = Var1, y = Var2, fill = value)) +  
 geom\_tile() + labs(title = "Correlation Heatmap")+  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))  
 }  
 })  
   
 filtered\_data <- reactive({  
 df\_decade[df\_decade$decade == input$decade\_slider,]  
 })  
   
 output$scatter\_plot <- renderPlot({  
 ggplot(filtered\_data(), aes(  
 x = !!input$scatter\_varX,   
 y = !!input$scatter\_varY)) +  
 geom\_point() +  
 ggtitle("Scatter Plot (Grouped by Decade)") +  
 xlim(0, 100) + ylim(0, input$y\_lim\_slider) +  
 theme\_minimal() +  
 if(input$smooth) {geom\_smooth()}  
 })  
   
 currentData <- reactive({input$country  
 })  
   
 output$timePlot <- renderPlot({  
 data <- wide\_narm[wide\_narm$country == input$country, ]  
   
 gdp <- ggplot(data, aes(x = year, y = gdppc\_constant)) +  
 geom\_line() +  
 ggtitle(paste("GDP per capita Over Time for", input$country)) +  
 ylab("GDP") + xlab("Year")  
   
 gni <- ggplot(data, aes(x = year, y = gnipc\_constant)) +  
 geom\_line() +  
 ggtitle(paste("GNI per capita Over Time for", input$country)) +  
 ylab("GNI") + xlab("Year")  
   
 world\_plot <- ggplot() +  
 geom\_polygon(data = world\_map, aes(x = long, y = lat, group = group),  
 fill = ifelse(world\_map$region == input$country, "red", "lightgray"),  
 color = "black") +  
 coord\_equal() +  
 labs(title = "World Map") + xlab(NULL) + ylab(NULL) +  
 theme\_light()  
   
 # Adjust plot size by specifying dimensions in plotOutput  
 plot\_grid(gdp, gni, world\_plot, ncol = 1, rel\_heights = c(2, 2, 3.5))  
 }, height = 800) # Specify desired height for the plot  
   
 }  
  
  
shinyApp(ui, server)