world_bank_project

Nick McCulloch, Cody Meagher, Stefano Musetti

2023-06-23

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

# 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

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 x 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)" "Lite
##  $ 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)
```

```r
# 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

```r
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
```

```r
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

```r
#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
```

```r
paste("max observations: ", max(year_filt$Freq))
```

```
## [1] "max observations:  258"
```

```r
paste("min observations: ", min(year_filt$Freq))
```

```
## [1] "min observations:  118"
```

```r
country_filt <- data.frame(table(gdp_filtered$country))

head(country_filt$Freq)
```

```
## [1] 20 62 62 42 62 20
```

```r
paste("max observations: ", max(country_filt$Freq))
```

```
## [1] "max observations:  62"
```

```r
paste("min observations: ", min(country_filt$Freq))
```

```
## [1] "min observations:  1"
```

```r
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
```

```r
#creating list of new countries from new data set.
new_countries <- read_csv("country_list_no_regions.csv")
```

```
## Rows: 217 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Country_Name
## dbl (1): Index
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
length(new_countries$Country_Name) #217
```

```
## [1] 217
```

```r
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

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

```
## [1] "factor"
```

```r
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.

```r
#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

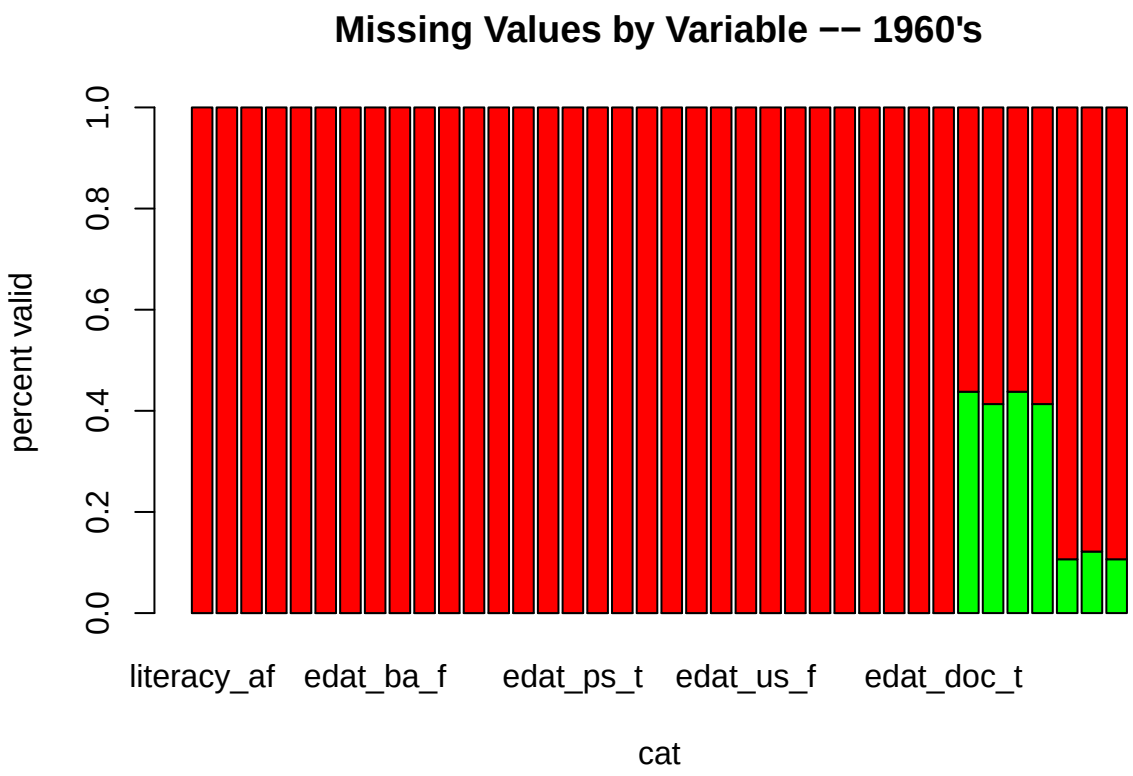
Creating df to be used later in shiny

```
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
```

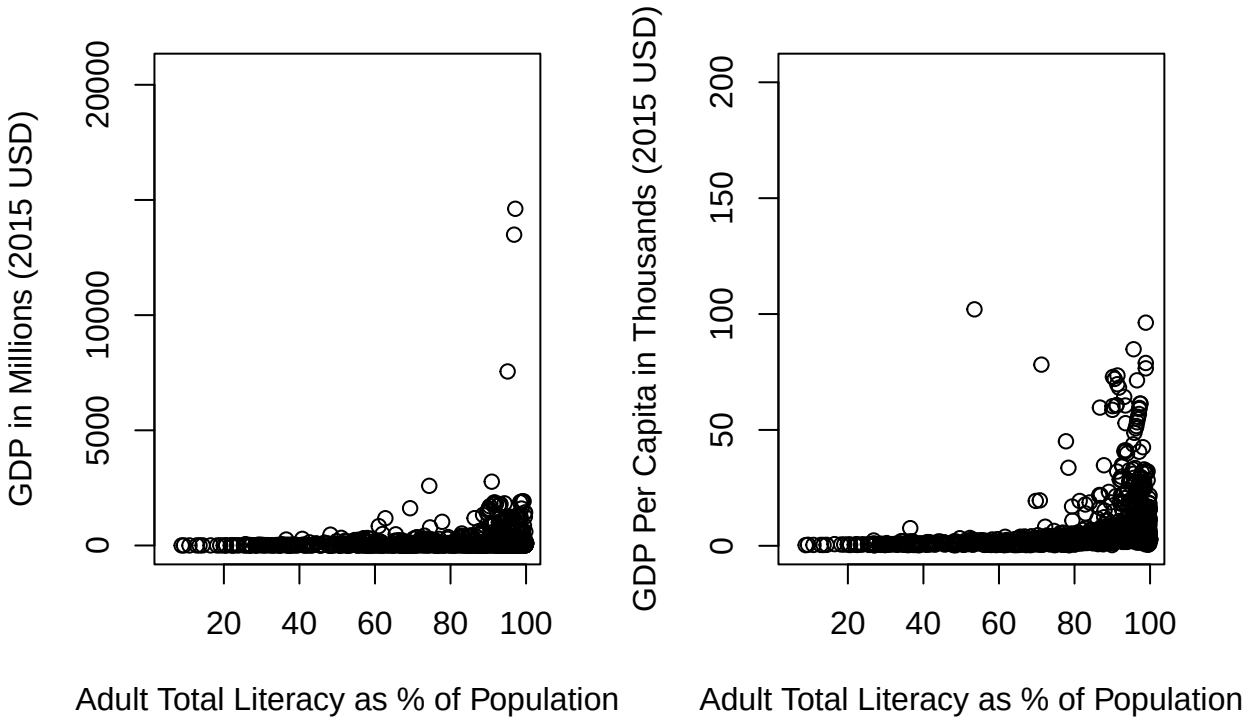
The remaining columns are wbdt: a long format data set, wbdt_wide: the wide version of wbdt, wb_narm: wbdt where all rows with NA's have been removed (less impactful in this case because each field has its own row), and wide_narm: where only rows with NA's in gnipc_constant have been removed.

Exploratory Analysis

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

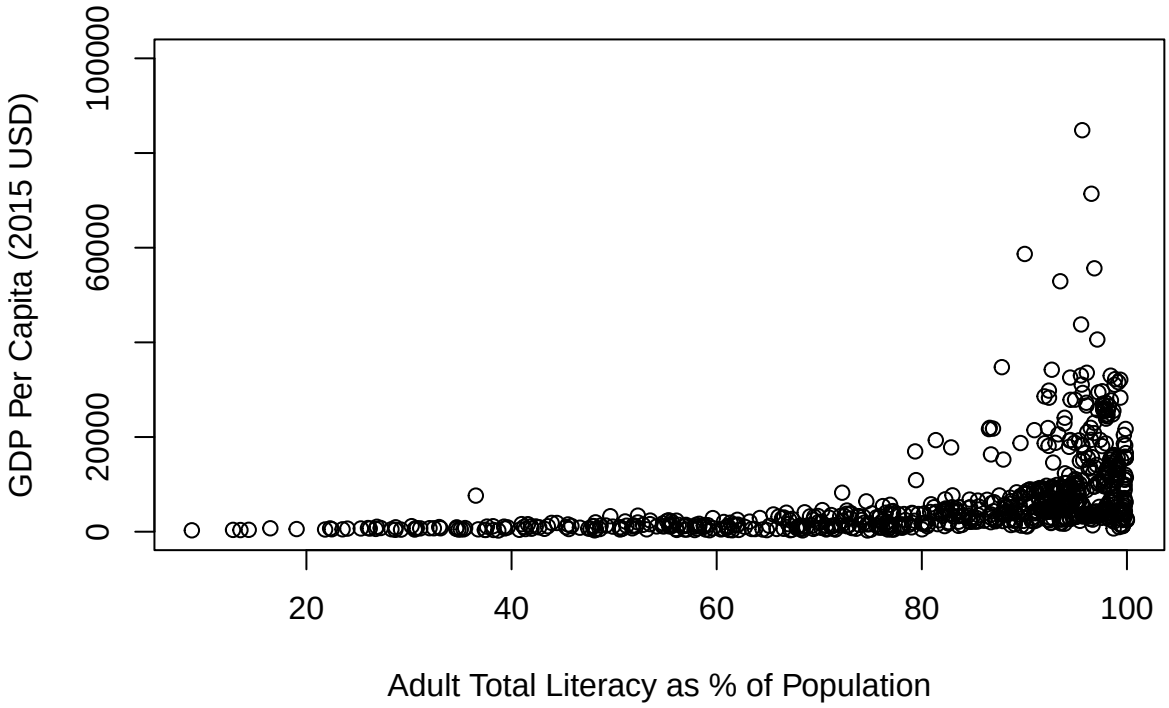
quick test plots

```r
par(mfrow = c(1,2))

plot(wbdt_wide$literacy_at, wbdt_wide$gdp_constant/1000000000, ylab = "GDP in Millions (2015 USD)", xlab

plot(x = wbdt_wide$literacy_at, y = wbdt_wide$gdppc_constant/1000, ylab = "GDP Per Capita in Thousands
```



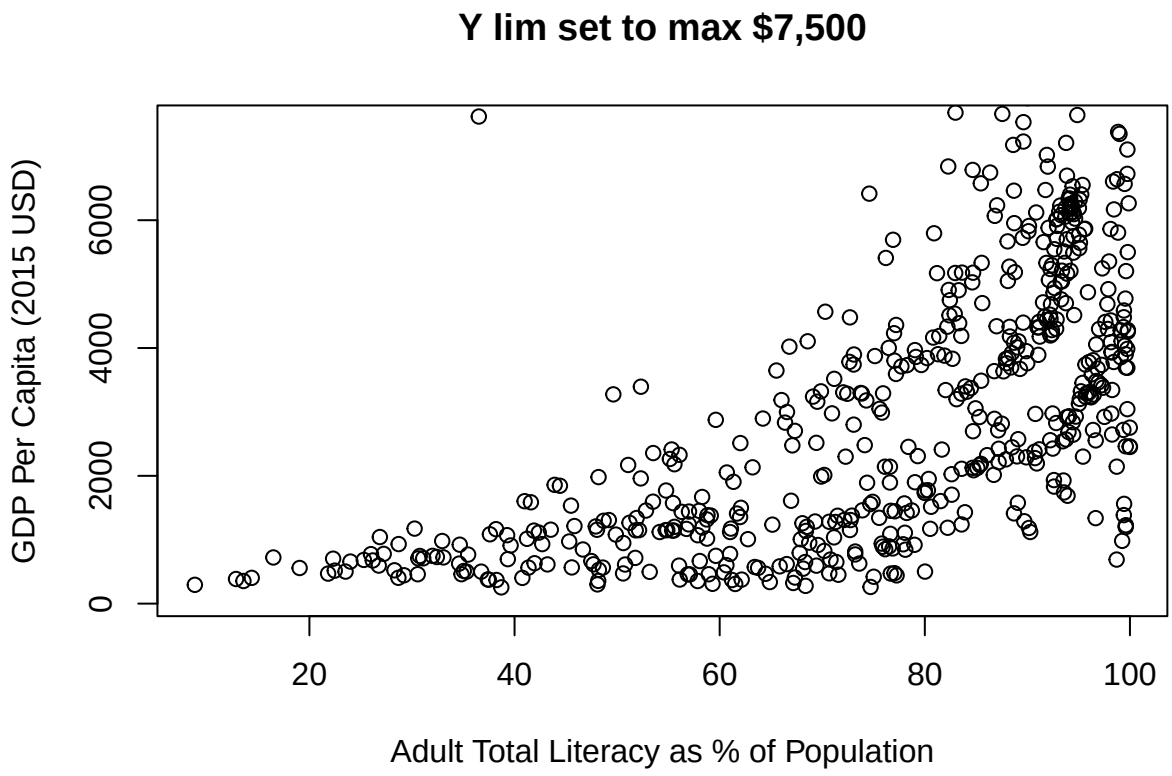
The plot seems to have significant outliers making it difficult to read.

```r
# 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
```

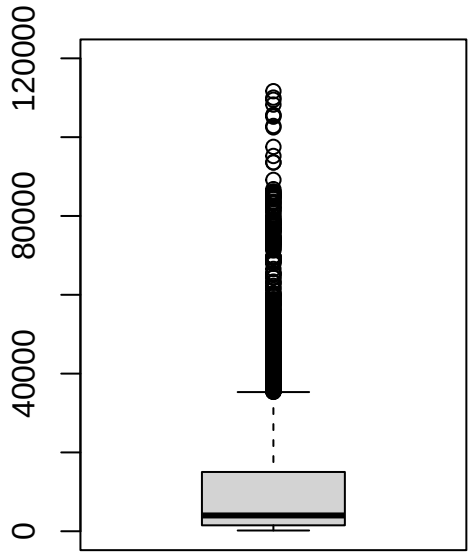


Although somewhat improved, the outliers are still a problem.

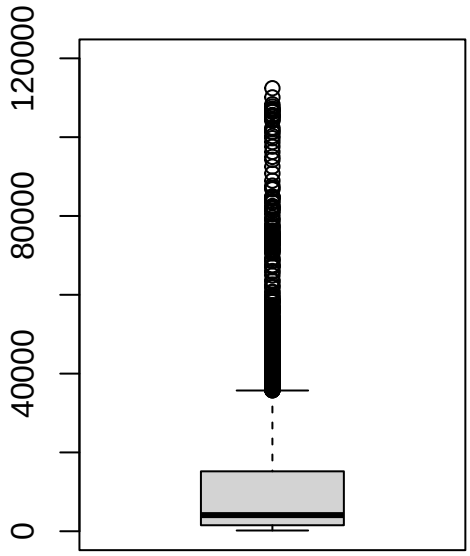
```
# recreating plot, limiting money range to 100/75K min/max
plot(wide_narm$literacy_at, wide_narm$gdppc_constant, ylab = "GDP Per Capita (2015 USD)", xlab = "Adult
```




```
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")
```



GNI per capita



GDP per capita

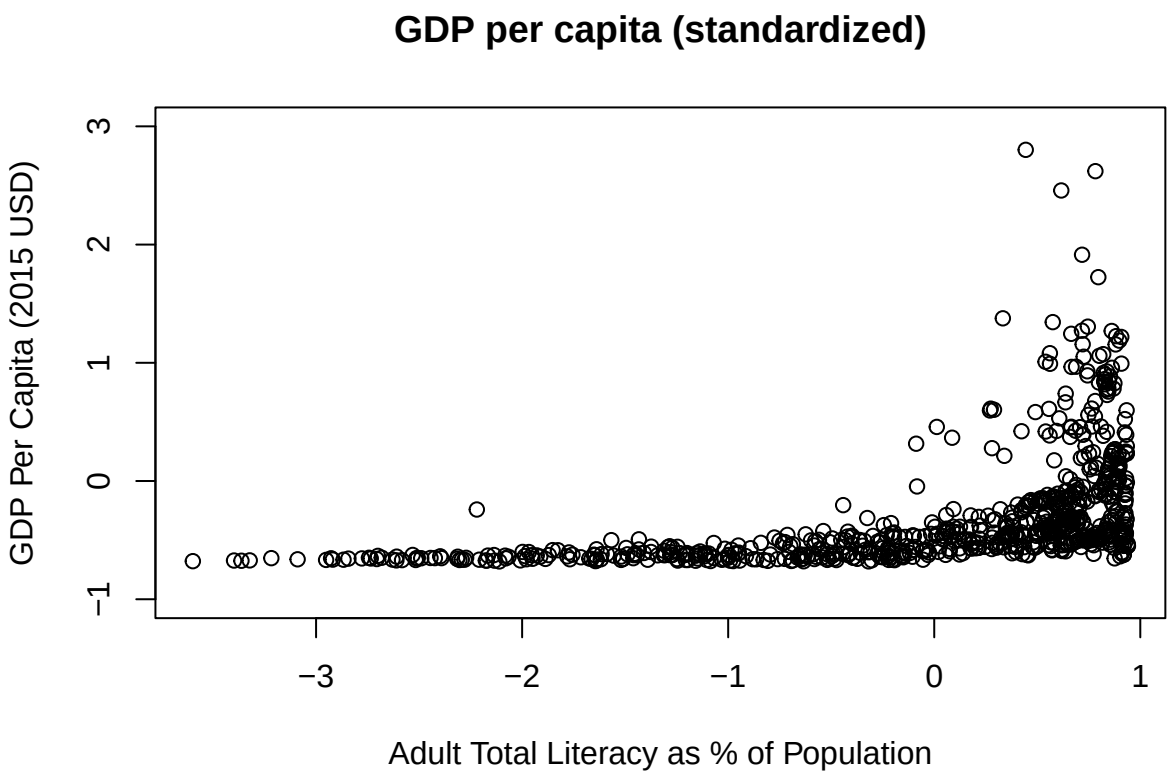
box plots of GDP and GNI indicate the same pattern even with limits set on y.

standardizing data

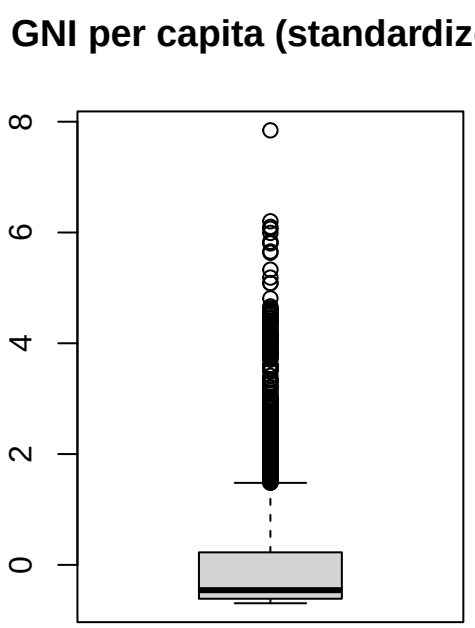
Standardizing the data highlights just how far from the norm the outliers are.

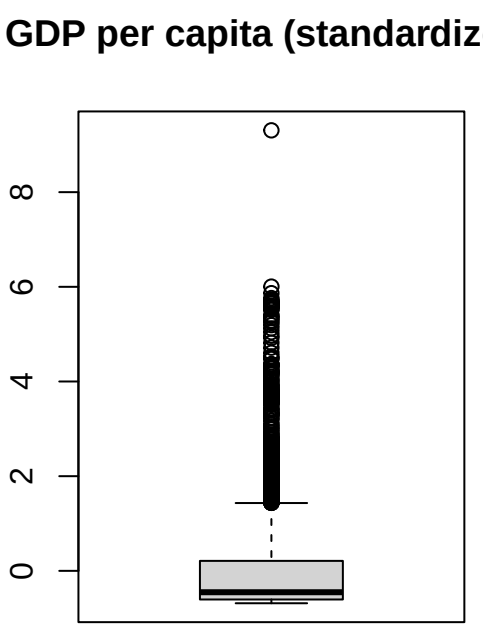
```
z_wide <- as.data.frame(scale(wide_narm[,!c("country","year")]))

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
```




```
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

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)
```

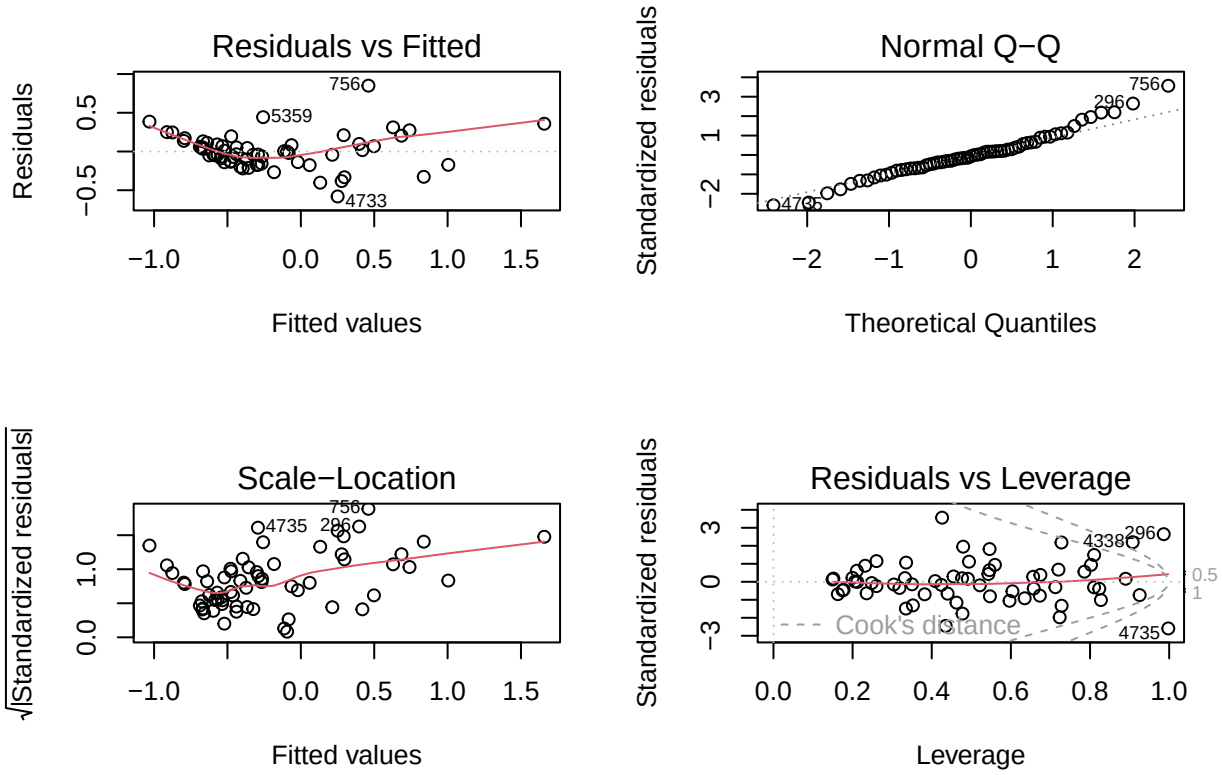
```
##
## Call:
## lm(formula = gnipc_constant ~ ., data = prelim_df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5802 -0.1348 -0.0190  0.0948  0.8497
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (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
```

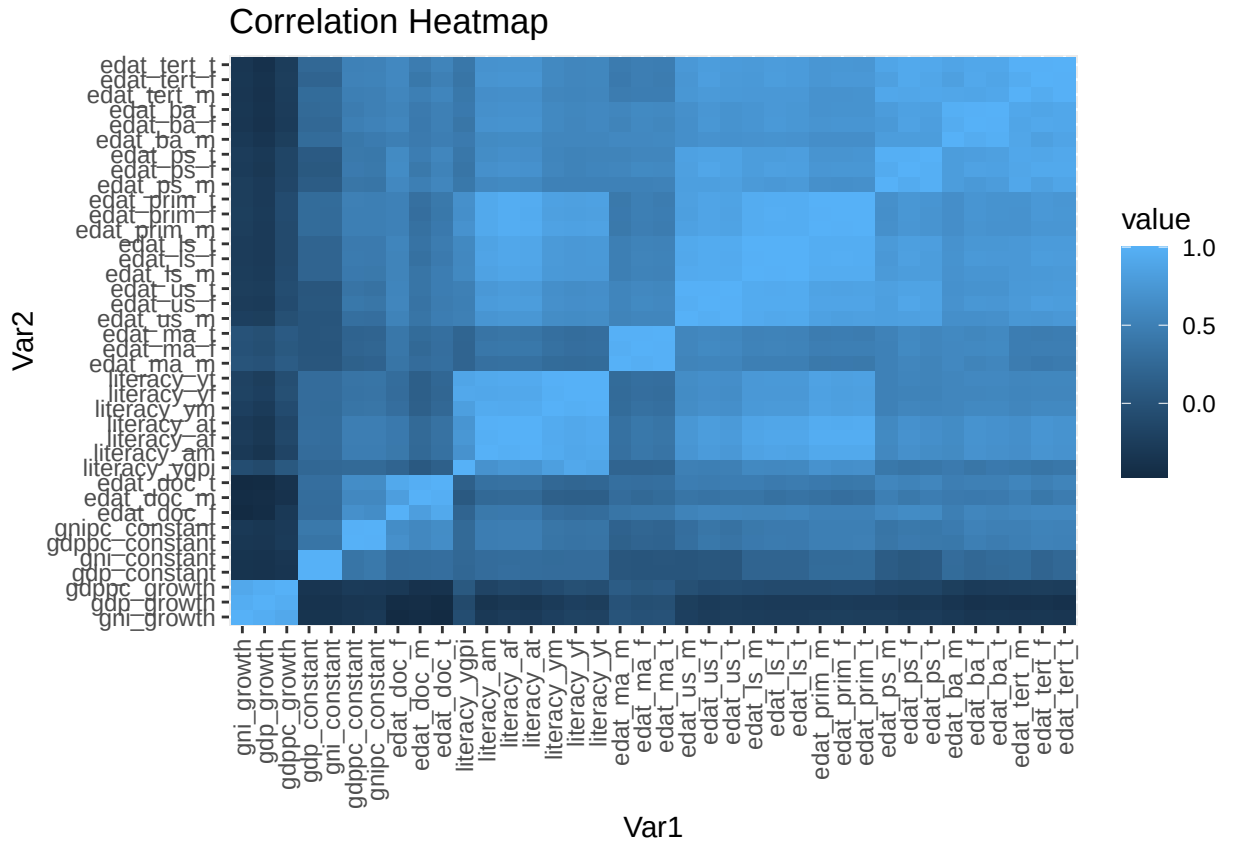
```
bptest(lm_prelim2, data = prelim_df2)
```

```
##
##  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 = "pea
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

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)
```

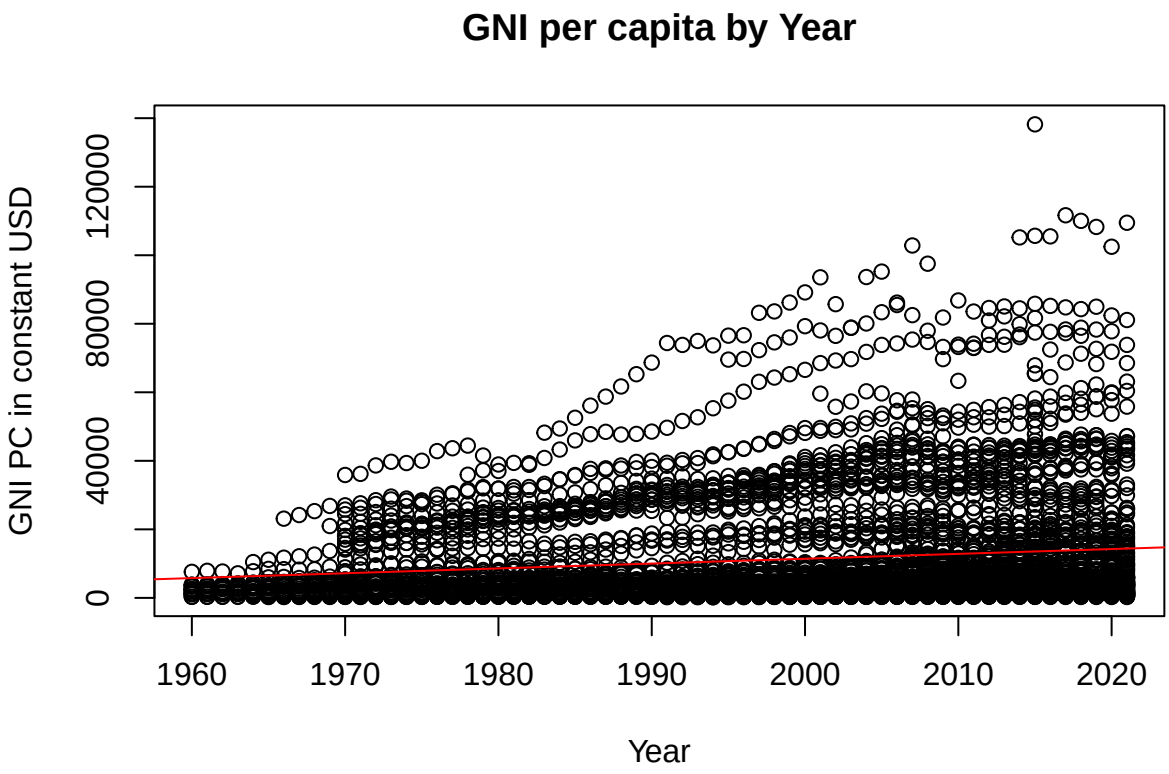
```
##
## Call:
## lm(formula = gnipc_constant ~ edat_us_t + edat_us_f + edat_us_m +
##     edat_tert_t, data = prelim_df1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.082 -0.485 -0.182  0.635  3.086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    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

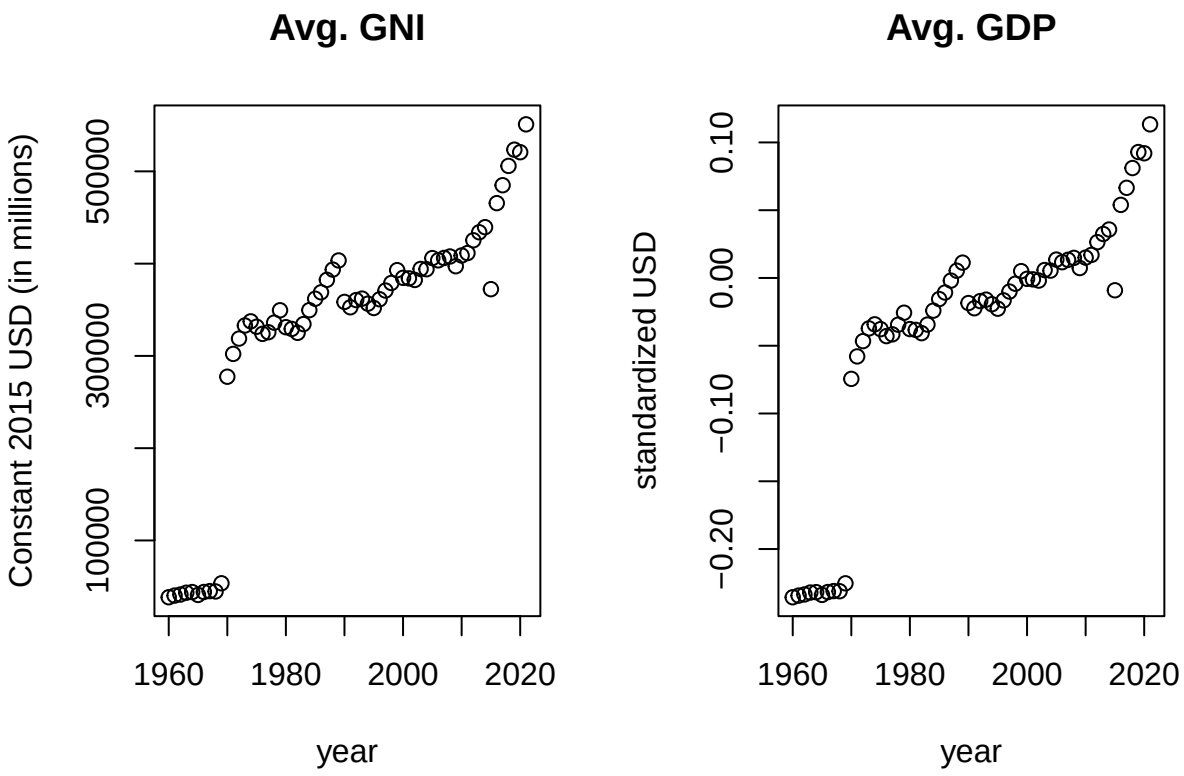
GNI_pc by year wtih fitted line

```
plot(wide_narm$year, wide_narm$gnipc_constant, main="GNI per capita by Year", xlab="Year", ylab="GNI PC
fit <- lm(gnipc_constant ~ year, data = wide_narm)
abline(fit, col = "red")
```



mean GNI and GDP by year

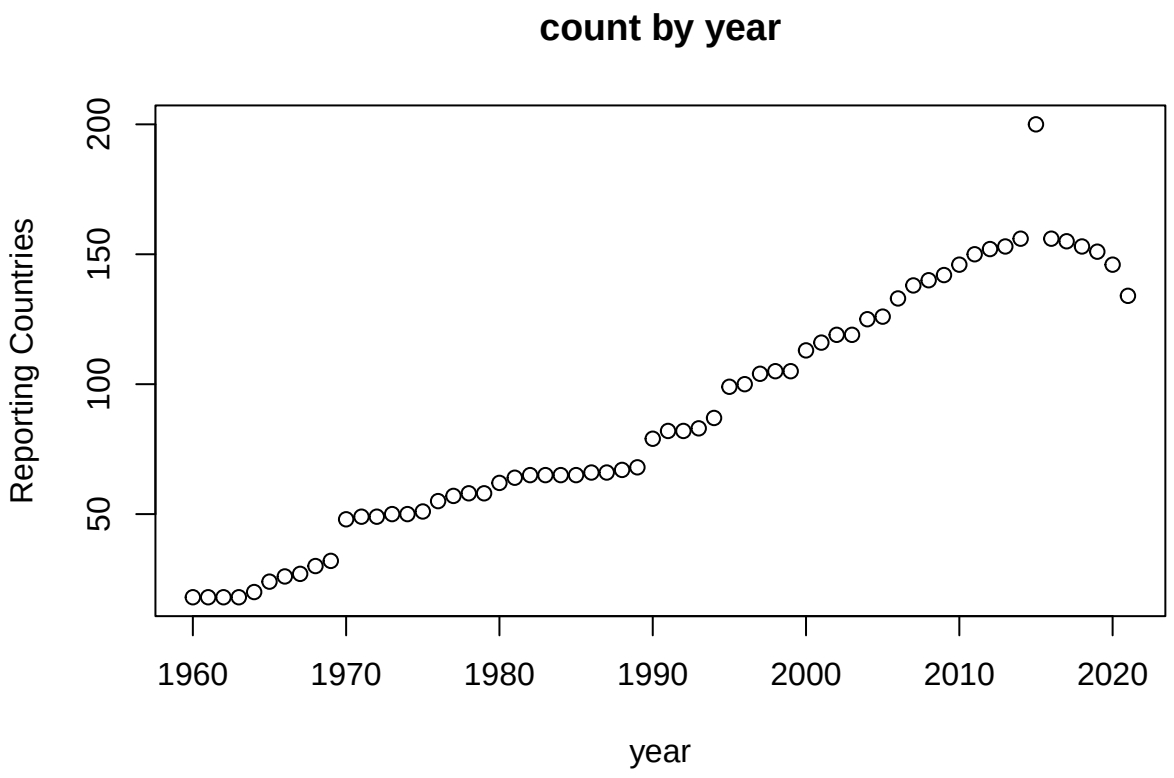
```
by_year <- wide_narm %>%
  group_by(year) %>%
  summarise(avg = mean(gni_constant/1000000))

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.

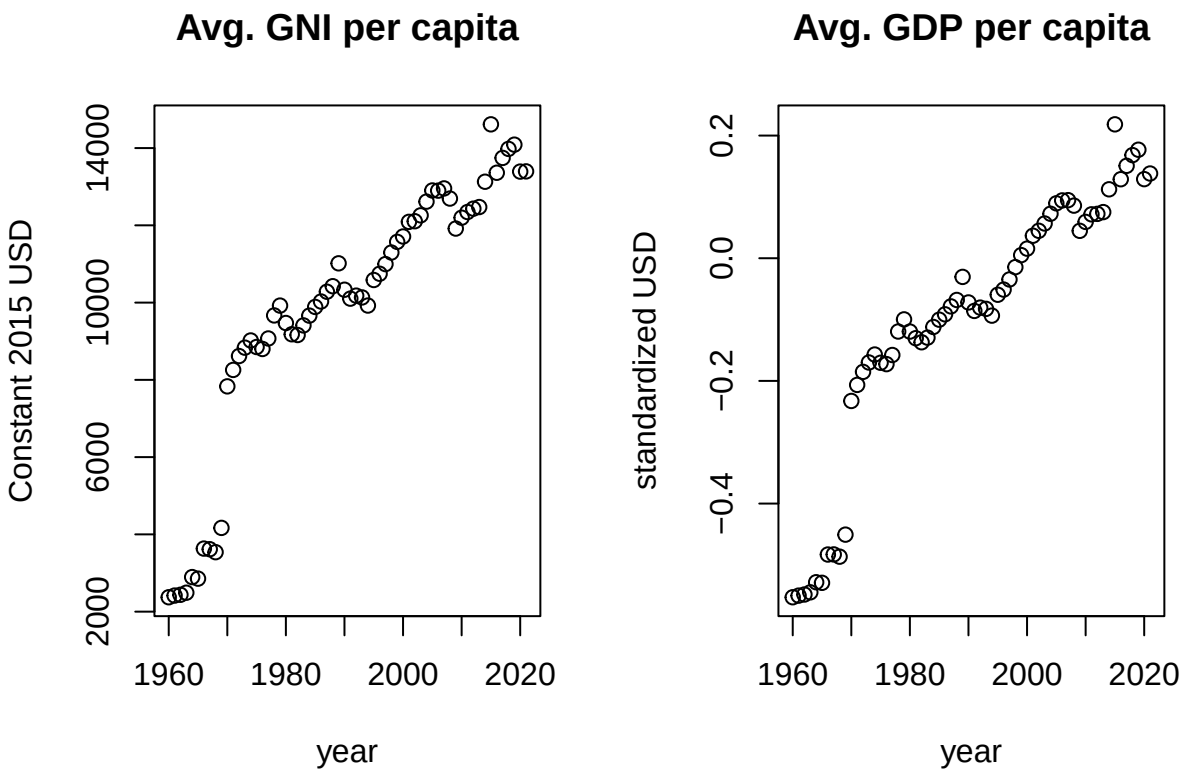
```
count_byyear <- wide_narm %>%
  group_by(year) %>%
  summarise(across(everything(), ~ sum(!is.na(.))))

plot(count_byyear$year,count_byyear$gni_constant, ylab = "Reporting Countries", main = "count by year",
```



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

A shiny app for the project was created using the code below and can be found here.

map data for shiny

The instructions at Sharp Sight 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.
```

```r
# printing list of country names in wold_map
world_map %>%
  group_by(region) %>%
  summarise() %>%
  print(n = Inf)
```

```
## # A tibble: 252 x 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
```

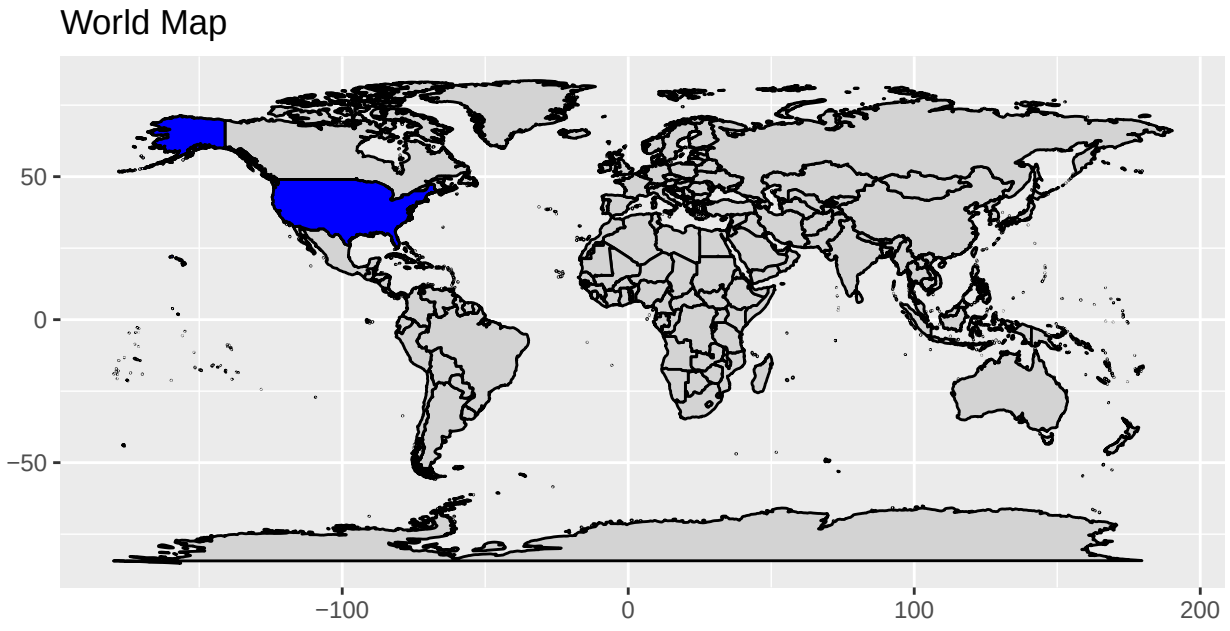
```
## 122 Lebanon
## 123 Lesotho
## 124 Liberia
## 125 Libya
## 126 Liechtenstein
## 127 Lithuania
## 128 Luxembourg
## 129 Madagascar
## 130 Madeira Islands
## 131 Malawi
## 132 Malaysia
## 133 Maldives
## 134 Mali
## 135 Malta
## 136 Marshall Islands
## 137 Martinique
## 138 Mauritania
## 139 Mauritius
## 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 = "bla
  geom_polygon(data = subset(world_map, region == "USA"), aes(x = long, y = lat, group = group), fill =
  coord_equal() +
  labs(title = "World Map")+xlab(NULL)+ylab(NULL)

world_plot
```



full RShiny code

```
library(pacman)

pacman::p_load(shinythemes, readr, dplyr, tidyverse, data.table, knitr, lmtest, lubridate, ggplot2, gri

library(mapdata)
```

```r
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")

# Group data by decade
df_decade <- df %>%
  mutate(decade = 10 * floor(year / 10))

resNum <- c(1,2,3,4,5,6)
resName <- c("Residuals vs Fitted",
             "Normal Q-Q",
             "Scale-Location",
             "Cook's Distance",
             "Residuals vs Leverage",
             "Correlation Heatmap")
residual_list <- setNames(as.list(resNum), resName)

ui <- fluidPage(theme = shinythemes::shinytheme("superhero"),
  titlePanel("World Bank Project"),
  fluidRow(
    mainPanel(h4("Group: Nick McCulloch, Cody Meagher, Stefano Mesetti"
    ))
  ),

  hr(),


  fluidRow(
    mainPanel(
      h4("Introduction"),),
    mainPanel("This project examined education and economic data from
              the World Bank's Development Indicator's Database.
              The Database contains information by country, year, and topic,
              covering everything from healthcare to criminal justice
              and every year from 1960 to the present.")
    ),

  fluidRow(
    mainPanel(
      h4("Data Limitations"
      )),
```

```
  mainPanel("As can be seen the dataset was notably sparse."),
    img(src="gif.gif", align = "left",height='450px',width='900px')
),

hr(),

fluidRow(
  sidebarLayout(
  sidebarPanel(
    selectInput("residual_var",
                label = "Select plot",
                choices = c(resName),
                selected = "Residuals vs Leverage"
    )
    ),

  mainPanel(
    plotOutput("residual_plot")
  ),
  )
),

hr(),

fluidRow(
  mainPanel(
    h4("Ed and Econ Plots"
    )),
),

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(
    mainPanel(h4("GNI and GDP Over Time"))),
  fluidRow(
      mainPanel(
        selectInput("country",
                    label = "Choose a country",
                    choices = unique(wide_narm$country),
                    selected = "South Africa"
        )
      ),

      mainPanel(
                              plotOutput("timePlot"),
                              plotOutput("countryplot"))
        )
)

server <- function(input, output) {
  output$residual_plot <- renderPlot({
    if(residual_list[[input$residual_var]] != 6) {
      plot(lm_prelim1, which = residual_list[[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()

    plot_grid(gdp, gni, world_plot, ncol = 1, rel_heights = c(2, 2, 3.5))
  }, height = 800)

  }

shinyApp(ui, server)
```