Data Analysis, Week of Feb 28

Harris Policy Lab

3/2/2021

This document explains some of the progress the NDI team made on data analysis from Feb 26 - March 2.

The datasets referenced in this file are on the [team GitHub](https://github.com/evantrowbridge/policy_lab_ndi).

## Controls for COVID-19 Outcomes

This week, we put together a dataframe of data for each country for the chosen control variables, using the most recent data from each country:

* GDP per capita
  + Latest year in data: 2019
* Gini index
  + Latest year in data: 2020
* Median age of population
  + Latest year in data: uncertain
* Portion of population age 65 or older
  + Latest year in data: uncertain
* Public health expenditure per capita
  + Latest year in data: 2018

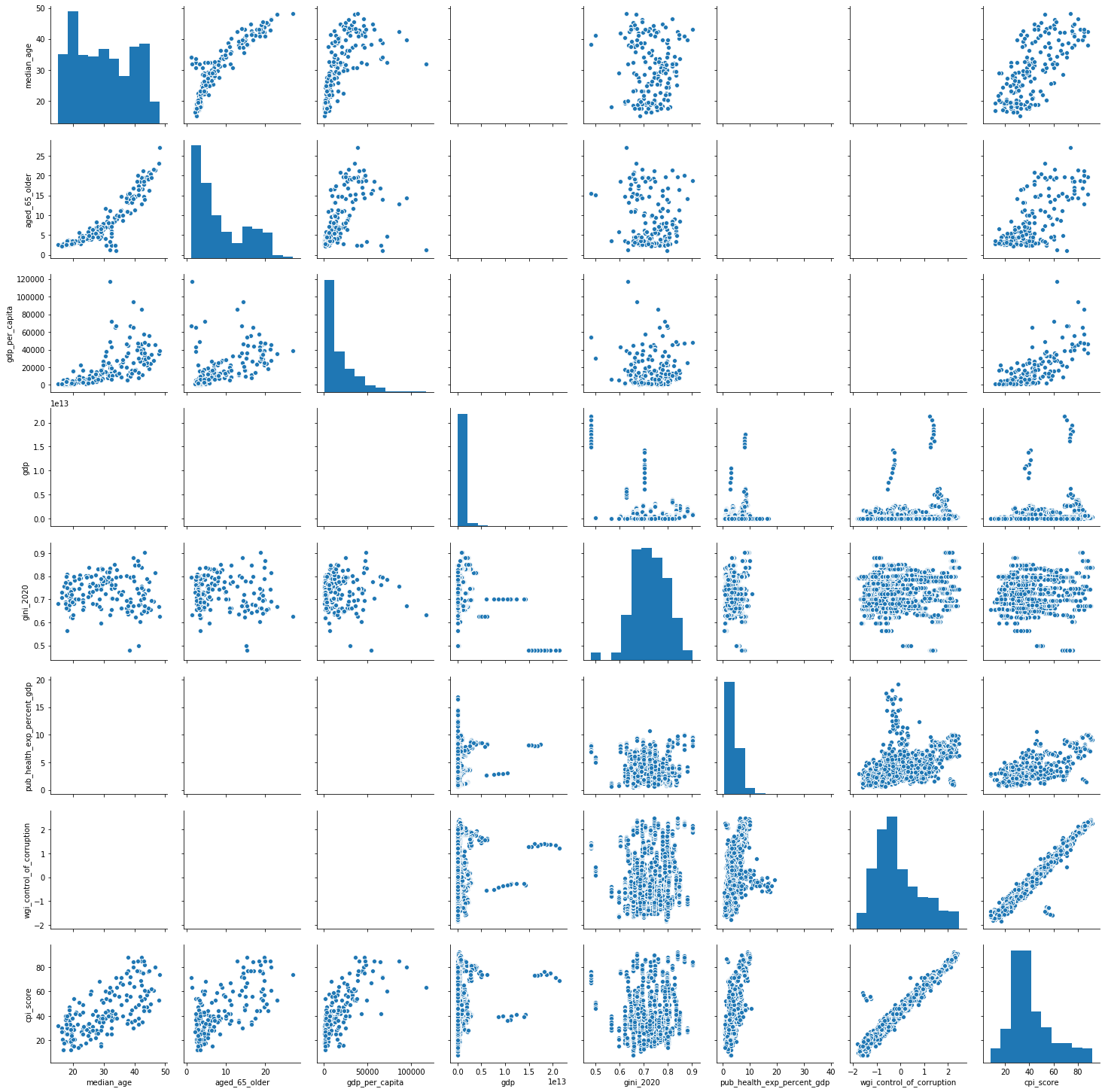
We have discussed including “Public health expenditure as percent of GDP,” but it seemed so similar to “Public health expenditure per capita” that I have it commented out in the code below. *How to think about this?*

The code to produce the controls is below.

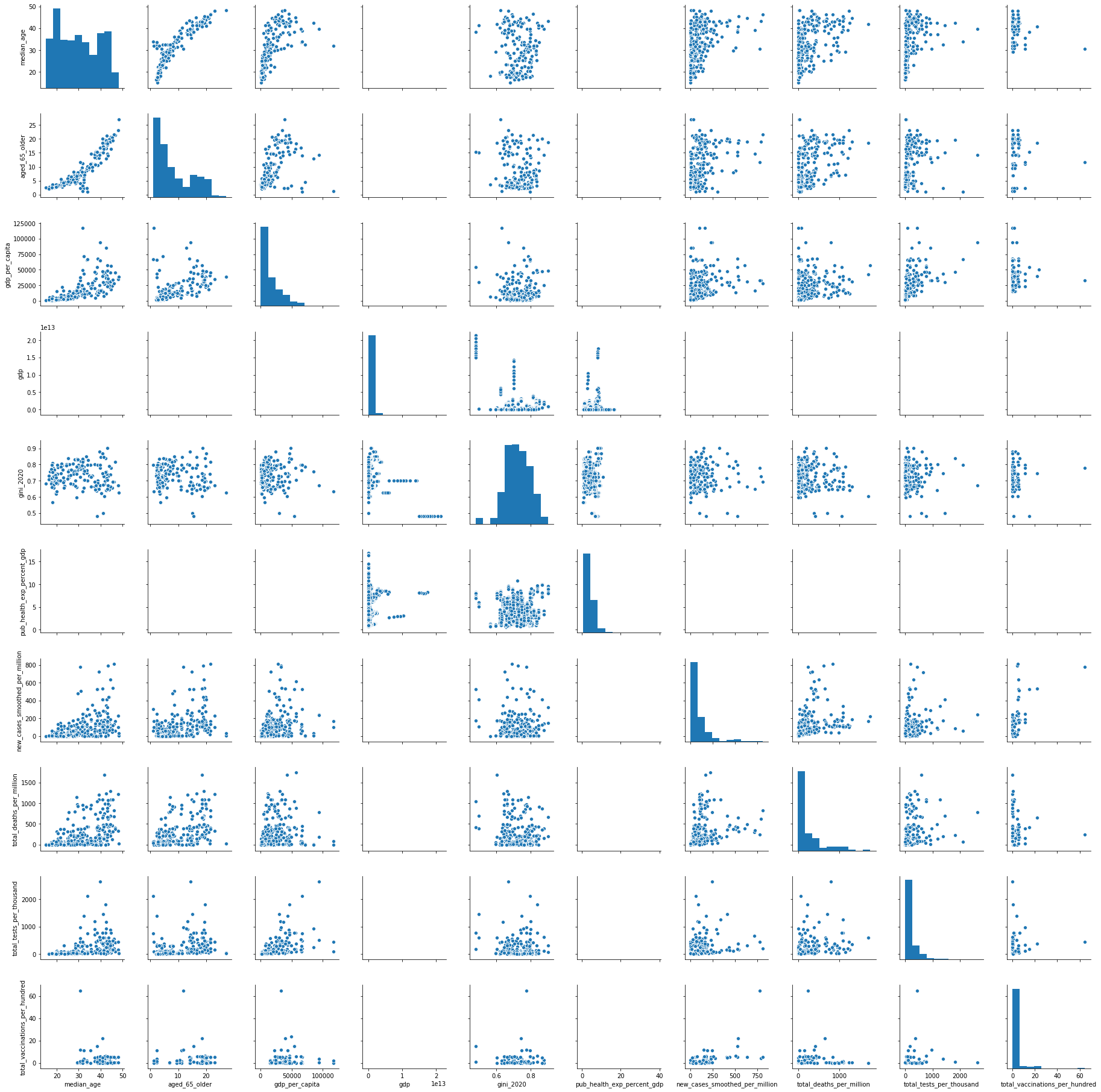
df\_all\_yr <- read\_csv("~/GitHub/policy\_lab\_ndi/data/merged\_data\_yr.csv")  
df\_all\_yr\_2020 <- read\_csv("~/GitHub/policy\_lab\_ndi/data/merged\_data\_yr\_2020.csv")  
  
# Controls for COVID outcomes  
df\_covid\_controls <- df\_all\_yr %>%  
 arrange(country\_standard, year) %>%  
 group\_by(country\_standard) %>%  
 # These variables don't have values in 2020. Filling in from most recent year  
 fill(  
 c(gdp, pub\_health\_exp\_percent\_gdp, percap\_domestic\_health\_expenditure)) %>%  
 filter(year==2020) %>%  
 select(  
 country\_standard,   
 gdp,   
 pub\_health\_exp\_percent\_gdp,   
 percap\_domestic\_health\_expenditure)  
   
df\_covid\_controls <- df\_all\_yr\_2020 %>%  
 select(  
 country\_standard, gdp\_per\_capita, gini\_2020, median\_age, aged\_65\_older) %>%  
 full\_join(df\_covid\_controls, by = "country\_standard") %>%   
 select(  
 country\_standard, gdp, gdp\_per\_capita, gini\_2020,   
 # we discussed using "public health expenditure as a proportion of GDP"  
 # as an indicator, but it seemed very similar to the next indicator  
 # below, which is "public health expenditure per capita"  
 #pub\_health\_exp\_percent\_gdp,   
 percap\_domestic\_health\_expenditure,   
 median\_age, aged\_65\_older)

We visualized how the controls relate with some variables in our data involving COVID-19, corruption, democracy, and transparency using pairplots. (The pairplots were made using differently-prepared control data, but compliments our understanding of the data).

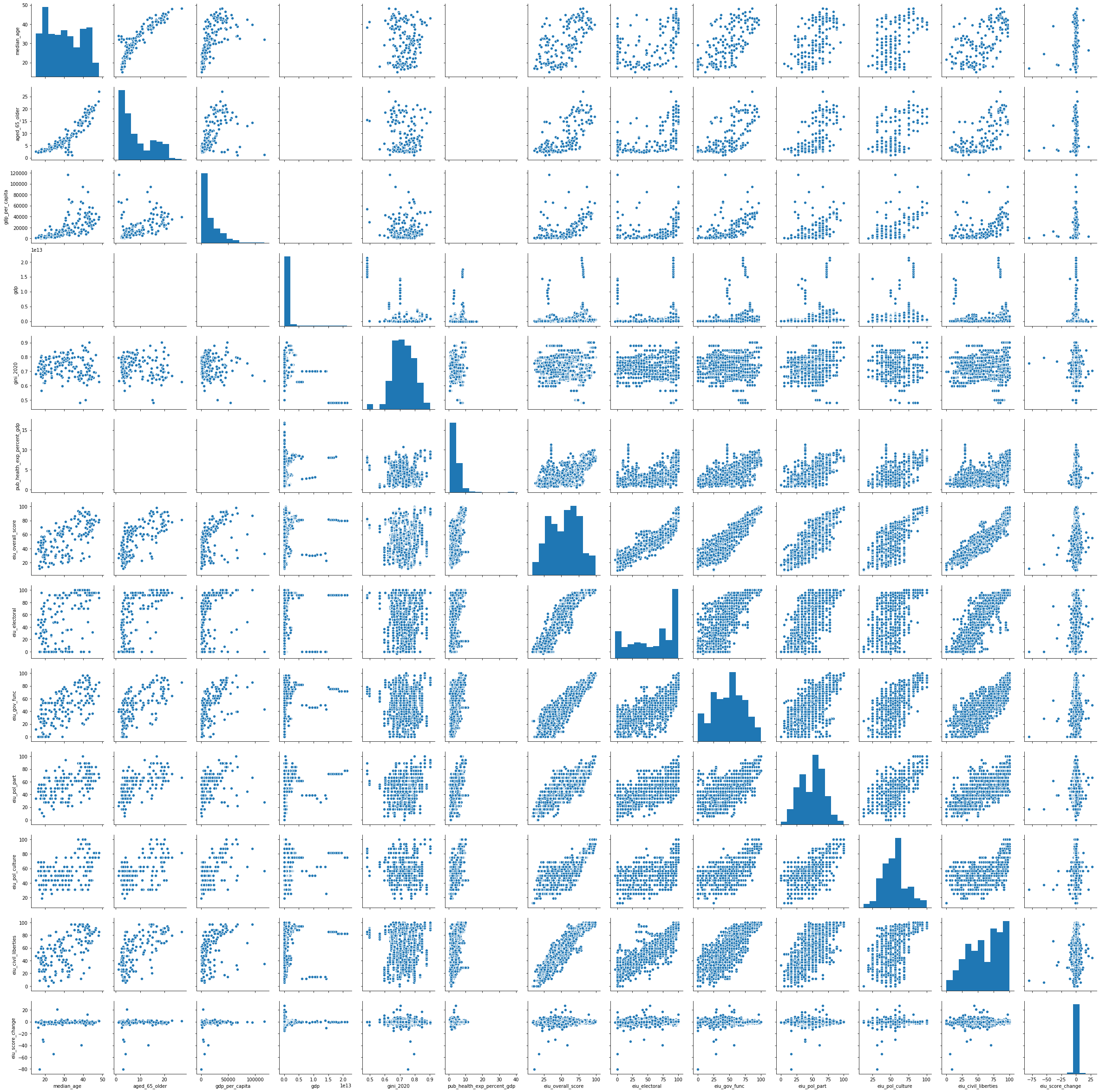
*Is it an endogeneity concern that there seems to be correlation between control variables and transparency variables?*



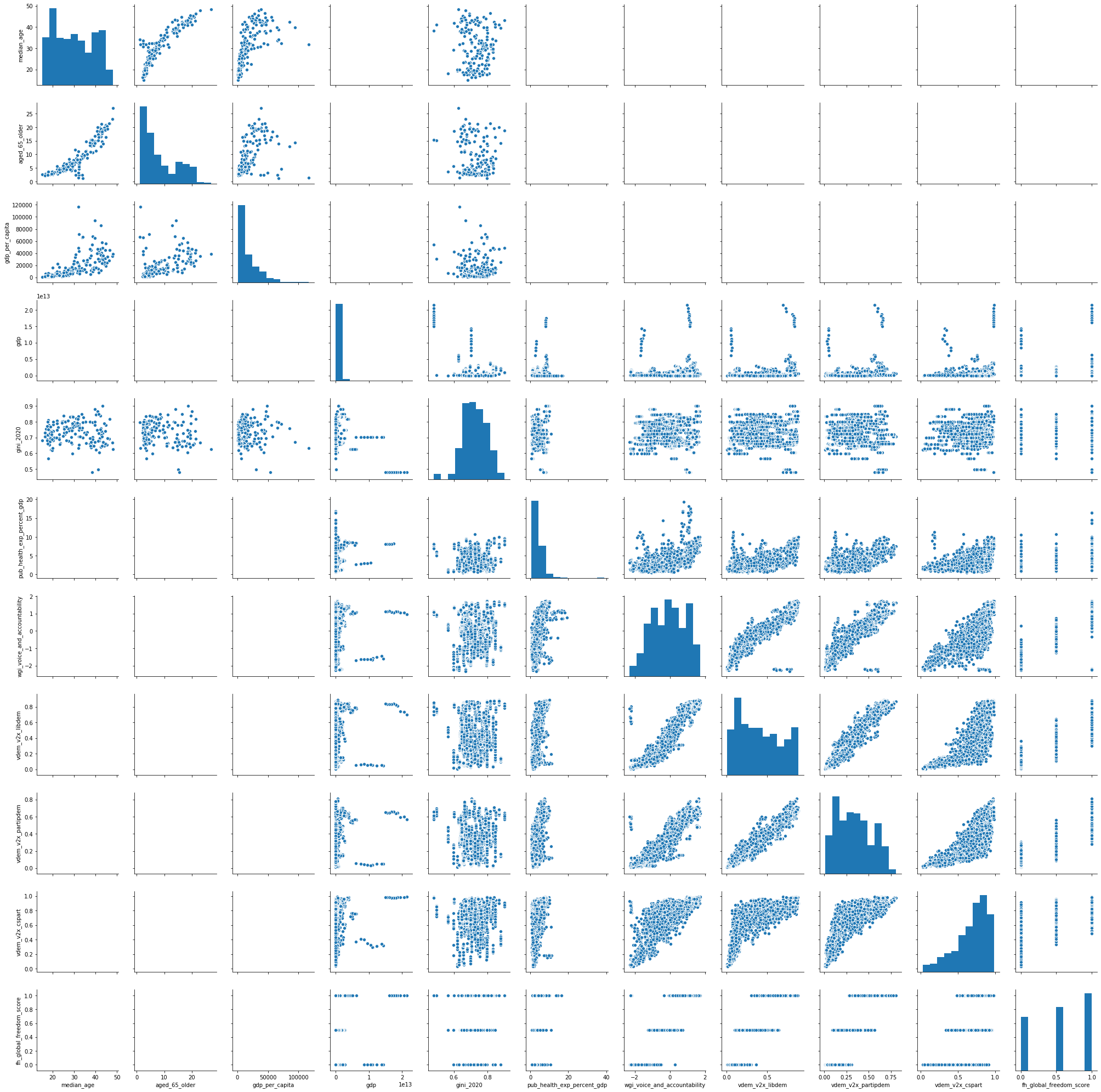
Corruption Indicators and Controls



COVID-19 Indicators and Controls



Democracy Indicators and Controls



Transparency Indicatos and Controls

## Creating an index for COVID-19 health factors

Our COVID-19 index comes from country-level measures starting from January 1, 2020. The measures are used in the following equation:

The intuition is that countries will have a higher index score if they have fewer cases, fewer deaths, and more tests. The index is scaled from 0 (worst) to 1 (best).

We decided on this index after testing it among six considered models. We regressed each index on our list of control variables and compared the F-statistic of each model. The chosen model had the second highest F-Statistic, but was chosen because a review of the country rankings of the model with the highest F-statistic revealed it to poorly reflect the reality of comparative outcomes across countries.

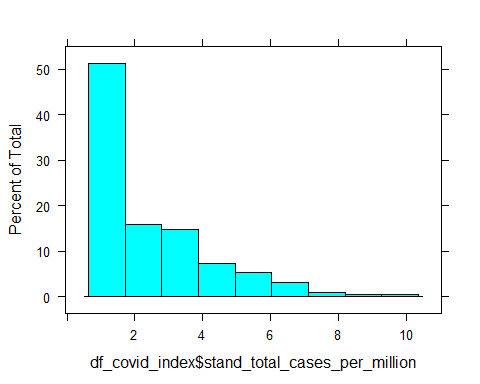
To see the code for the process above, see [this portion of the group’s GitHub](https://github.com/evantrowbridge/policy_lab_ndi/blob/main/Reference_Code/covid_index_select.r).

The process for making this index can be adapted for the remaining indices our group will be creating.

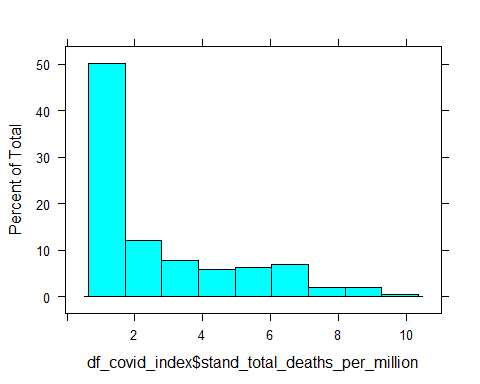
For countries that were missing data on COVID-19 tests, they are assigned a score of “1” on the scale of 1 (lowest performing) to 10 (highest performing) used for countries’ testing rates.

The code to create the index is below:

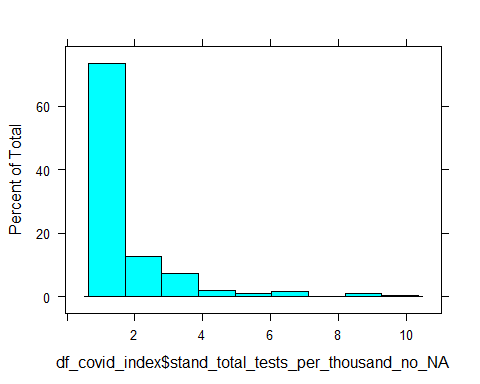
df\_covid\_index <- df\_all\_yr %>%  
 select(  
 country\_standard,   
 cum\_total\_cases\_per\_million,   
 cum\_total\_deaths\_per\_million,  
 cum\_total\_tests\_per\_thousand) %>%  
 filter(  
 if\_any(  
 cum\_total\_cases\_per\_million:cum\_total\_deaths\_per\_million,   
 ~ !is.na(.))) %>%   
 distinct() %>%  
 mutate(  
 # Use a scale of 1 - 10 to avoid having "0" in denominator  
 stand\_total\_cases\_per\_million = rescale(cum\_total\_cases\_per\_million, to = c(1, 10)),  
 stand\_total\_deaths\_per\_million = rescale(cum\_total\_deaths\_per\_million, to = c(1, 10)),  
 stand\_total\_tests\_per\_thousand = rescale(cum\_total\_tests\_per\_thousand, to = c(1, 10)),  
 stand\_total\_tests\_per\_thousand\_no\_NA = ifelse(  
 is.na(stand\_total\_tests\_per\_thousand),   
 1,   
 stand\_total\_tests\_per\_thousand),  
 covid\_index = rescale(  
 (1/stand\_total\_cases\_per\_million) +  
 (1/stand\_total\_deaths\_per\_million) +  
 stand\_total\_tests\_per\_thousand\_no\_NA)  
 ) %>%  
 select(country\_standard, covid\_index, everything())  
  
# Reviewing distribution of the data  
histogram(df\_covid\_index$stand\_total\_cases\_per\_million)



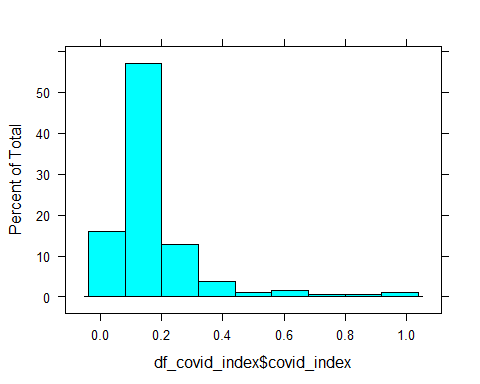
histogram(df\_covid\_index$stand\_total\_deaths\_per\_million)



histogram(df\_covid\_index$stand\_total\_tests\_per\_thousand\_no\_NA)



histogram(df\_covid\_index$covid\_index)



# RUnning a regression of control variables on COVID-19 index  
covid\_index\_mod\_test <- df\_covid\_index %>%   
 select(country\_standard, covid\_index) %>%  
 full\_join(df\_covid\_controls, by = "country\_standard") %>%  
 select(-country\_standard)  
  
 lm(covid\_index ~ ., data = covid\_index\_mod\_test) %>% summary()

##   
## Call:  
## lm(formula = covid\_index ~ ., data = covid\_index\_mod\_test)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.34923 -0.06285 0.00081 0.03053 0.61423   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.574e-01 1.275e-01 2.803 0.00573 \*\*  
## gdp -5.506e-15 5.367e-15 -1.026 0.30652   
## gdp\_per\_capita 3.414e-06 1.068e-06 3.197 0.00169 \*\*  
## gini\_2020 -2.266e-01 1.578e-01 -1.436 0.15316   
## percap\_domestic\_health\_expenditure 2.867e-05 1.557e-05 1.842 0.06741 .   
## median\_age -1.851e-03 3.971e-03 -0.466 0.64180   
## aged\_65\_older -3.558e-03 5.721e-03 -0.622 0.53491   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1299 on 151 degrees of freedom  
## (36 observations deleted due to missingness)  
## Multiple R-squared: 0.2895, Adjusted R-squared: 0.2613   
## F-statistic: 10.26 on 6 and 151 DF, p-value: 1.64e-09

## Creating an Index for Corruption Indicators

The index for corruption indicators combines Transparency International’s Corruption Perception Index (CPI) and the World Bank Worldwide Governance Indicator’s Control of Corruption (CoC) index.

It standardizes both indicators on a scale of 0 (worse) to 1 (best).

It then creates two options for indices:

Option One is a sum of the two indicators:

Option Two is the product of the two indicators:

data <- df\_all\_yr  
  
#transform wgi from factor to numeric  
data2 <- data  
data2$wgi\_voice\_and\_accountability <- as.character(data$wgi\_voice\_and\_accountability)  
data2$wgi\_voice\_and\_accountability <- as.numeric(data2$wgi\_voice\_and\_accountability)  
data2$wgi\_control\_of\_corruption <- as.character(data$wgi\_control\_of\_corruption)  
data2$wgi\_control\_of\_corruption <- as.numeric(data2$wgi\_control\_of\_corruption)  
class(data2$wgi\_control\_of\_corruption)

## [1] "numeric"

#checking each variable  
summary(data2$cpi\_score)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 8.00 33.00 34.00 36.54 37.00 92.00 3223

summary(data2$wgi\_control\_of\_corruption)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## -1.869 -0.749 -0.233 -0.003 0.725 2.470 8754

#rescaling and creating index of corruption (cpi\_score: 0-100, WGI: -2.5~2.5)  
corruption\_index\_pre <- data2 %>%  
 mutate(data2, cpi\_score\_scale = data2$cpi\_score/100) %>%  
 mutate(data2, wgi\_coc\_scale = (data2$wgi\_control\_of\_corruption+2.5)/5) %>%  
 mutate(data2, corruption\_idx\_sum = ifelse(year >= 2012, cpi\_score\_scale + wgi\_coc\_scale, NA )) %>%  
 mutate(data2, corruption\_idx\_avg = ifelse(year >= 2012, (cpi\_score\_scale + wgi\_coc\_scale)/2, NA ))  
  
#checking rescaled variable and index  
summary(corruption\_index\_pre$cpi\_score)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 8.00 33.00 34.00 36.54 37.00 92.00 3223

summary(corruption\_index\_pre$wgi\_control\_of\_corruption)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## -1.869 -0.749 -0.233 -0.003 0.725 2.470 8754

summary(corruption\_index\_pre$cpi\_score\_scale)

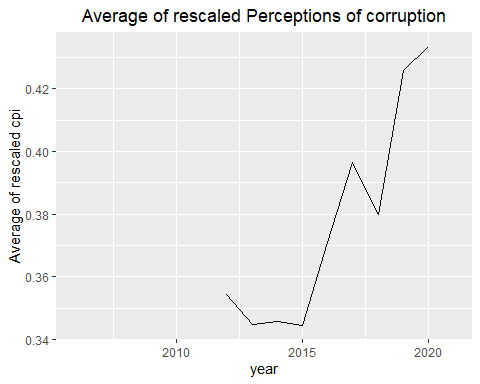
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.080 0.330 0.340 0.365 0.370 0.920 3223

summary(corruption\_index\_pre$wgi\_coc\_scale)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.126 0.350 0.453 0.499 0.645 0.994 8754

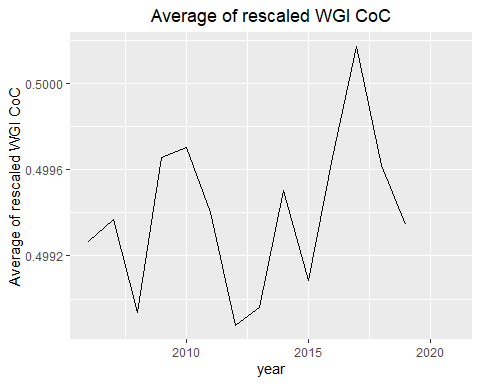
#create csv file of index  
write.csv(corruption\_index\_pre[c(1:2, 109:110)], "Corruption index.csv")  
  
#plotting rescaled variables  
avg\_cpi\_score\_scale <- corruption\_index\_pre %>%  
 group\_by(year) %>%  
 summarise(mean\_cpi\_scale = mean(cpi\_score\_scale, na.rm = TRUE))  
  
ggplot(data = avg\_cpi\_score\_scale, aes(x=year, y =mean\_cpi\_scale)) +  
geom\_line() + labs(title = "Average of rescaled Perceptions of corruption", y = "Average of rescaled cpi") +  
theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 7 row(s) containing missing values (geom\_path).



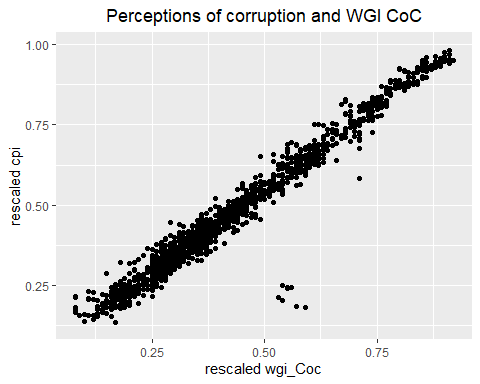
avg\_wgicoc\_score\_scale <- corruption\_index\_pre %>%  
 group\_by(year) %>%  
 summarise(mean\_wgicoc\_scale = mean(wgi\_coc\_scale, na.rm = TRUE))  
  
ggplot(data = avg\_wgicoc\_score\_scale, aes(x=year, y =mean\_wgicoc\_scale)) +  
 geom\_line() + labs(title = "Average of rescaled WGI CoC", y = "Average of rescaled WGI CoC") +  
 theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 2 row(s) containing missing values (geom\_path).



ggplot(data = corruption\_index\_pre, aes(x= cpi\_score\_scale, y = wgi\_coc\_scale)) +  
 geom\_point() + labs(title = "Perceptions of corruption and WGI CoC", x = "rescaled wgi\_Coc", y = "rescaled cpi") +  
 theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 10291 rows containing missing values (geom\_point).



#cheking index  
summary(corruption\_index\_pre$corruption\_idx\_sum)

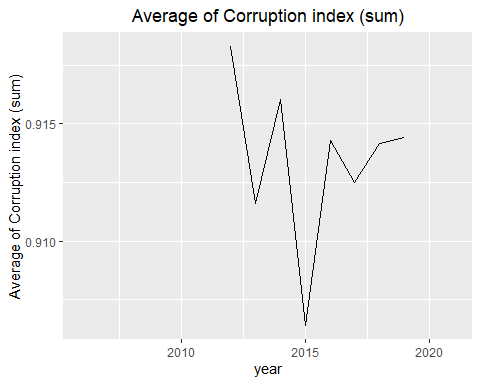
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.240 0.626 0.806 0.914 1.187 1.891 10291

summary(corruption\_index\_pre$corruption\_idx\_avg)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.120 0.313 0.403 0.457 0.594 0.945 10291

#plotting index  
avg\_corruption\_idx\_sum <- corruption\_index\_pre %>%  
 group\_by(year) %>%  
 summarise(mean\_corruption\_idx\_sum = mean(corruption\_idx\_sum, na.rm = TRUE))  
  
ggplot(data = avg\_corruption\_idx\_sum, aes(x=year, y =mean\_corruption\_idx\_sum)) +  
 geom\_line() + labs(title = "Average of Corruption index (sum)", y = "Average of Corruption index (sum)") +  
 theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 8 row(s) containing missing values (geom\_path).



avg\_corruption\_idx\_avg <- corruption\_index\_pre %>%  
 group\_by(year) %>%  
 summarise(mean\_corruption\_idx\_avg = mean(corruption\_idx\_avg, na.rm = TRUE))  
  
ggplot(data = avg\_corruption\_idx\_avg, aes(x=year, y =mean\_corruption\_idx\_avg)) +  
 geom\_line() + labs(title = "Average of Corruption index (avg)", y = "Average of Corruption index (avg)") +  
 theme(plot.title = element\_text(hjust = 0.5))

## Warning: Removed 8 row(s) containing missing values (geom\_path).

