Emerging Markets indices, GDP and population

An exploration into four emerging markets. This paper will explore the relationship between indices, GDP and population of four countries.

Even Oscar Harlert

2023-10-18

Introduction

Emerging markets can be great investment opportunity. In order to make good investment decisions and understand the risk and return for any investment you have to analyze historical data.

To measure countries development there are several metrics that can be useful. In this paper we will keep things simple and only use two metrics and that is gross domestic product (GDP onwards) and population. We could use a lot more metrics but to limit the scope of this paper we are only using GDP and population.

In total we are considering indices (including a benchmark of FTSE world), GDP and population. Even though the amount of data is small we hope to derive some interesting insights.

Research question

This paper will have a limited scope. We are seeking to investigate indices data representing each country. The countries selected for this paper are India, Egypt, China and Brazil. Our research questions will be the following:

- How has the historical development been in indices funds been in India, Egypt, China and Brazil?
- What has the historical risk been?
- Is there a relationship between GDP per capita and the indices development for each country?

In this paper, we consider risk to be standard deviation of the dataset for the indices.

Data sets

There are three datasets we have acquired for this paper. The first one is downloaded from refinitive/datastream. For more information regarding the data provider you can follow this link. This will be refereed to as dataset one. The data in the file are time-series spanning from 19. September 2003 to 19. September 2023. The measurements are done with a monthly interval. All data points are represented in US dollar.

The second data set is downloaded from The World Bank. The data-set is actually divide into two files but will be combined in RStudio as they each only contain one metric of data, GDP and population.

Project libraries

This project will make use of several libraries in RStudio. The following code snippet will load all the necessary libraries for this project.

```
library(here)
```

```
here() starts at D:/R_folder/Assignments/Emerging_markets
```

```
library(tidyverse)
```

```
Warning: package 'dplyr' was built under R version 4.3.2
```

```
Warning: package 'stringr' was built under R version 4.3.2
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```r
library(readxl)
library(psych)
```

```
Attaching package: 'psych'

The following objects are masked from 'package:ggplot2':

    %+%, alpha
```

```r
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(readxl)
library(ggplot2)
library(reshape2)
```

```
Warning: package 'reshape2' was built under R version 4.3.2
```

```
Attaching package: 'reshape2'

The following object is masked from 'package:tidyr':

    smiths
```

With all the libraries loaded we are ready to start working with our data sets. To be able to replicate the current sets, I have made a github repository where you can access the files. These can be accessed here. Dataset two is accessible from World Bank DataBank. However, the refinitive dataset would required you to have a refinitive account. Therefore, to assure that the project is reproducible I made the repository linked above.

In all, we have three excel files in the folder data sets that can be accessed in the project folder.

Import, tidy & transformation

This section will be split into three sections:

1. Import, tidy and transform dataset one
2. Import tidy and transform dataset two
3. We join the data-sets into data-frames we can use for visualization and modelling.

Dataset one

Initially we want to start to access the data we collected in the dataset folder. The data is stored as a excel file that has been stored in the project folder. To make it possible to work with in RStudio we start by importing the file and adding the data into a data-frame called index_emerging_markets.

```
index_emerging_markets <- read_excel("datasets/FTSE 20.09.23 - EOH.xlsx", skip = 6)

index_emerging_markets <- index_emerging_markets |>
  rename(
    Date = Name
  )
```

We changed to name of the column for dates to dates from names with the second row. The data in this file is currently in absolute values of the indices. To be able to make comparisons we need to normalize the data with the start of 100. In order to do so we are using the following:

```
normalized <- index_emerging_markets |>
  mutate(
    FTSEWorldN = FTSEWorld / (146.2 / 100),
    IndiaN = India / (332.5 / 100),
    EgyptN = Egypt / (59.04 / 100),
    ChinaN = China / (672.28 / 100),
    BrazilN = Brazil / (150.65 / 100),
  )
```

The code above adds 6 new columns where the data is normalized and starts at 100 on the 19. September 2003. The code snippet above can be generalized as the following:

Initial value / (Initial value / 100). The initial value divided by 100 will be exectued on each of the observation on each row. This results in a graph where we can compare each indicies including our benchmark (FTSE World).

To save some space we will combine a couple of lines of code:

```
normalized_trimmed <- normalized |>
  select(Date, FTSEWorldN:BrazilN)

normalized_trimmed <- normalized_trimmed |>
  mutate(
    FTSEWorld_Log = c(NA, log(FTSEWorldN[-1] / FTSEWorldN[-nrow(normalized)])),
    India_Log = c(NA, log(IndiaN[-1] / IndiaN[-nrow(normalized)])),
    EgyptN_Log = c(NA, log(EgyptN[-1] / EgyptN[-nrow(normalized)])),
    China_Log = c(NA, log(ChinaN[-1] / ChinaN[-nrow(normalized)])),
    Brazil_Log = c(NA, log(BrazilN[-1] / BrazilN[-nrow(normalized)])),
  )

normalized_trimmed <- normalized_trimmed |>
  mutate(
    FTSE_per = FTSEWorld_Log * 100,
    India_Per = India_Log * 100,
    Egpyt_per = EgyptN_Log * 100,
    China_per = China_Log * 100,
    Brazil_per = Brazil_Log * 100,
  )

log_return <- select(normalized_trimmed,
    -"FTSEWorldN":-"BrazilN"
  )

log_return <- log_return[-1, ]

desc_data <- describe(normalized_trimmed) |>
  t()
```

```
Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
```

```
Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

```
desc_data <- as.data.frame(desc_data)

desc_data <- desc_data[-c(1, 5:9), ]

desc_data <- select(desc_data,
```

```
                    -"FTSE_per":-"Brazil_per"
                    )

final_desc_data_log <- select(desc_data,
                              FTSEWorld_Log:Brazil_Log
                              )

write.csv(final_desc_data_log, "exported_data/desc_data_log.csv")
```

The lines above does several things, in short:

- We add rows for geometric returns (most commonly used while working with historical returns in finance).
- We create new data-frames that separates data.
- We use describe() (a function from the psych package), it returns descriptive statistics from the dataset and we store this in a new data-frame called desc_data.
- A few more selective operations followed by an export to a csv if we at a later point would like to use the descriptive data from the project.

At this point we have data-frames with the data organized to make further analysis later in the project.

As mentioned earlier we used a function called describe, it returns the data in a frame where we have some abbreviations, for anyone not familiar with statistics they are the following:

Table 1: Furthermore, all the descriptive data is contextual and even though it does not provide any direct information to answer our research questions it does provide information about the data set. The skew indicates if the return are aligning more on the left of or the right side of the distribution. The kurtosis are relevant due to indicating extreme values deviating from the mean. In financial terms this would be extreme profit or loss. The risk (standard deviation) and the mean (return) are directly related to the question.

Short	Definition
n	Number of observations
mean	The avarage
sd	Standard deviation (this is considered risk in financial terms)
skew	Skewness is the degree of asymmetry observed in a probability distribution (source).
kurtosis	An indicator of the tail of the distribution (source).
se	Standard error, and estimation of the standard deviation.

Dataset two

The second dataset is actually two excel files downloaded from the World Bank. Again, we need to load the data into a dataframe:

```
countries <- c("India", "Egypt, Arab Rep.", "China", "Brazil")
gdp <- read_excel("datasets/gdp.xls", skip = 3)
population <- read_excel("datasets/population.xls", skip = 3)
```

In addition to loading two dataframes with the data from the files, we also stored the four values (the names of the countries we are working with).

The data-sets are filled with data we do not have any use for therefore we need to tidy it:

```
gdp <- clean_names(gdp)
population <- clean_names(population)

gdp_clean <- select(gdp,
  -"x1960":-"x2002",
  -"indicator_code",
  -"indicator_name",
  -"country_code"
  ) |>
  filter(
    country_name %in% countries
  ) |>
  t() |>
  row_to_names(row_number = 1) |>
  as.data.frame()

population_clean <- select(population,
  -"x1960":-"x2002",
  -"indicator_code",
  -"indicator_name",
  -"country_code"
  ) |>
  filter(
    country_name %in% countries
  ) |>
  t() |>
  row_to_names(row_number = 1) |>
  as.data.frame()
```

```r
population_clean <- population_clean %>%
  rownames_to_column(var="Date")

population_clean$Date <- sub("^x", "", population_clean$Date)
names(population_clean)[4] <- "Egypt"

gdp_clean <- gdp_clean %>%
  rownames_to_column(var="Date")
gdp_clean$Date <- sub("^x", "", gdp_clean$Date)
names(gdp_clean)[4] <- "Egypt"

population_clean$Brazil <- as.numeric(population_clean$Brazil)
population_clean$China <- as.numeric(population_clean$China)
population_clean$Egypt <- as.numeric(population_clean$Egypt)
population_clean$India <- as.numeric(population_clean$India)
population_clean$Date <- as.numeric(population_clean$Date)

gdp_clean$Brazil <- as.numeric(gdp_clean$Brazil)
gdp_clean$China <- as.numeric(gdp_clean$China)
gdp_clean$Egypt <- as.numeric(gdp_clean$Egypt)
gdp_clean$India <- as.numeric(gdp_clean$India)
gdp_clean$Date <- as.numeric(gdp_clean$Date)

population_clean_long <- pivot_longer(population_clean, cols = -Date, names_to = "Country"
population_clean_long$Date <- as.numeric(population_clean_long$Date)

gdp_clean_long <- pivot_longer(gdp_clean, cols = -Date, names_to ="Country", values_to = "
gdp_clean_long$Date <- as.numeric(gdp_clean_long$Date)

gdp_pop <- left_join(population_clean_long, gdp_clean_long, by = c("Date", "Country"))
names(gdp_pop)[3] <- "Population"
names(gdp_pop)[4] <- "GDP"
```

Above we are simply cleaning out and making usable data frames for later in the project and for future visualization. We also assure that the data is in its correct form.

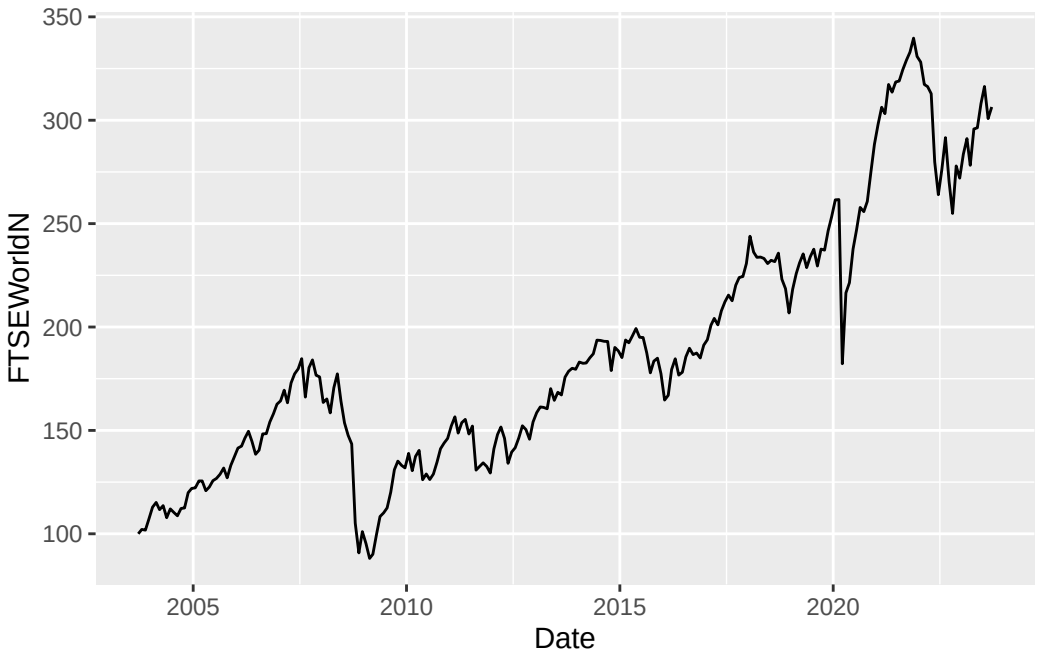
Visualization

Until now we only have several data-frames and descriptive statistics from the data-sets. In this section we will visualize the data to improve our understanding of what the data represent.

The first thing we want to get a better overview over is dataset one.

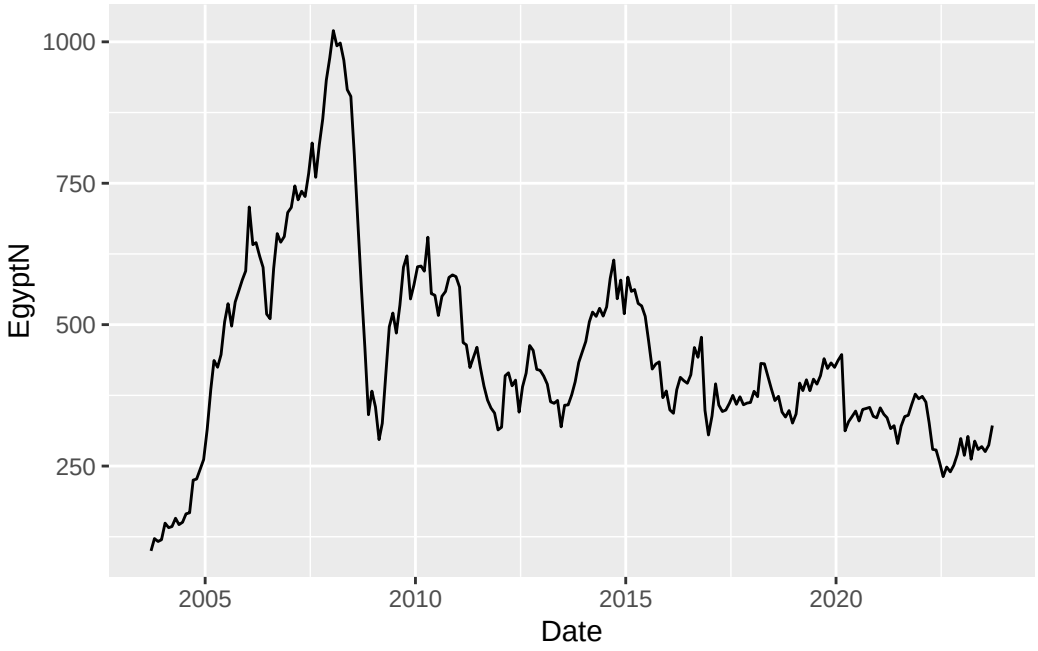
The normalized development of the FTSE Indices can simply be graph it out doing the following:

```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date, y = FTSEWorldN )
) +
  geom_line( mapping = aes())
```

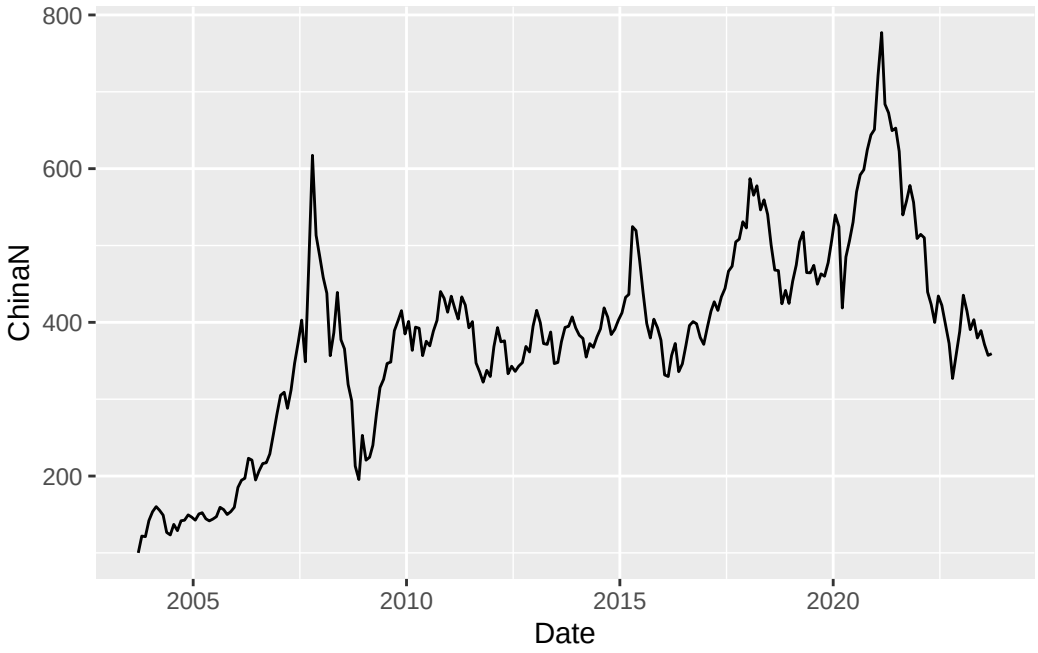



```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date, y = IndiaN )
) +
  geom_line( mapping = aes())
```

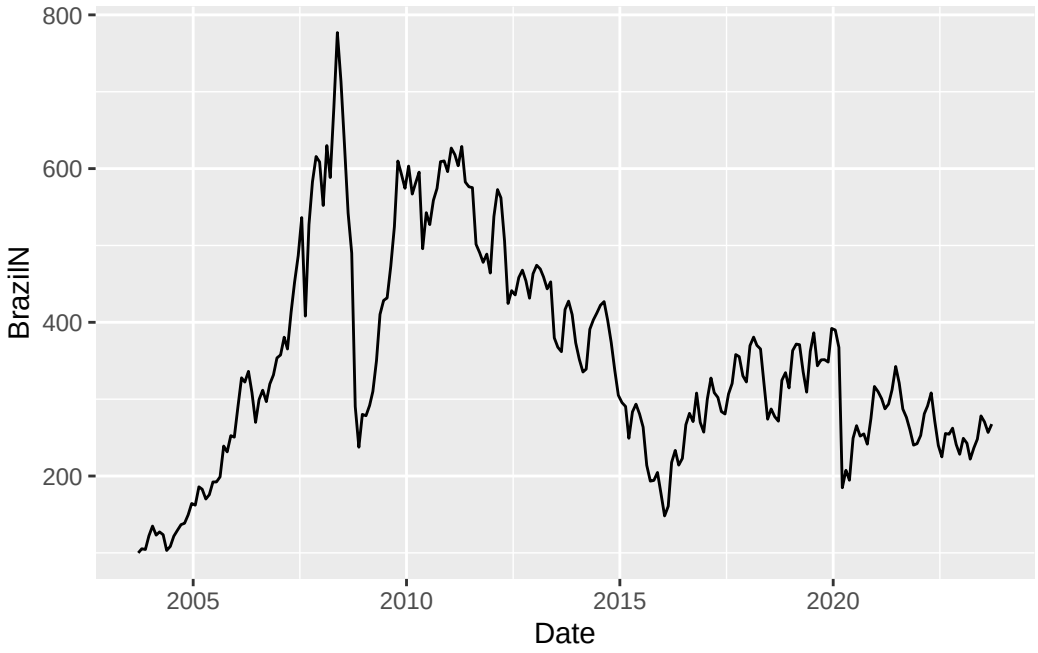

```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date, y = EgyptN )
) +
  geom_line( mapping = aes())
```




```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date, y = ChinaN )
) +
  geom_line( mapping = aes())
```

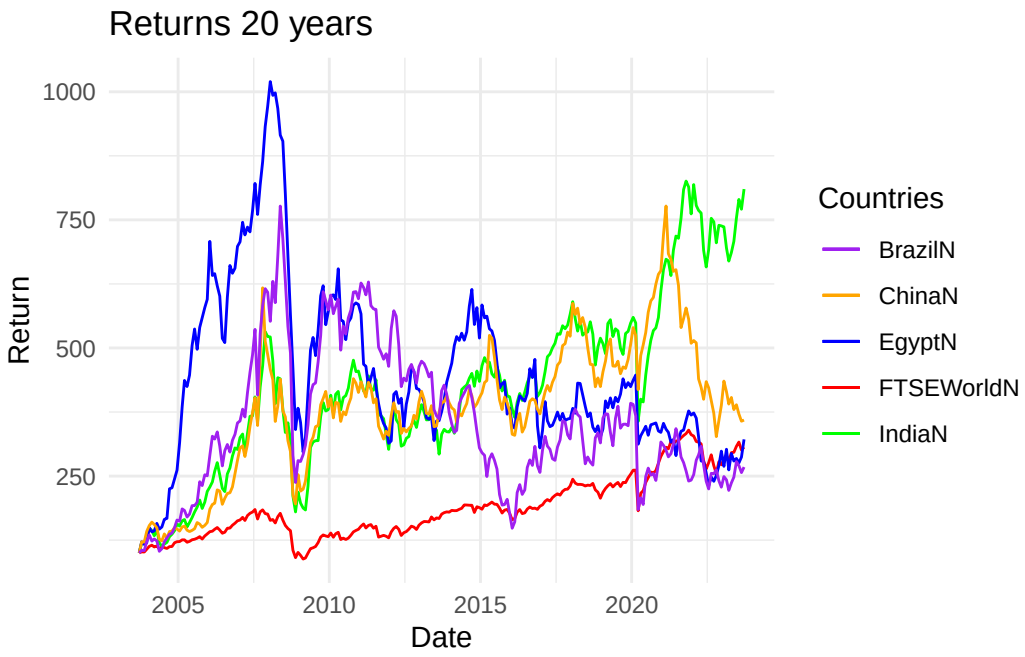



```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date, y = BrazilN )
) +
  geom_line( mapping = aes())
```



The returns combined in one image results in the following:

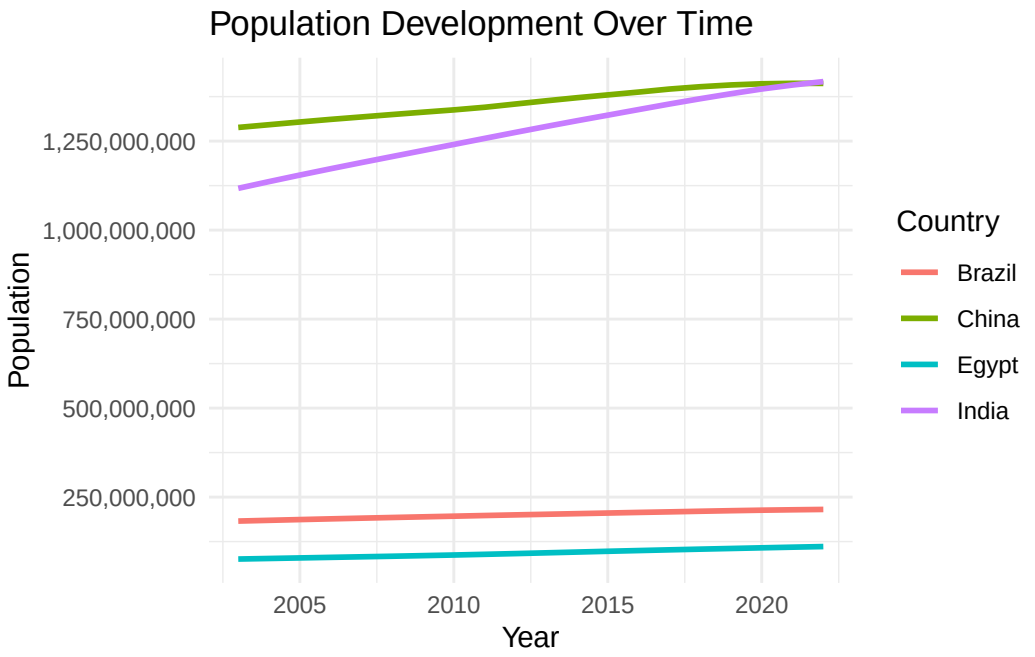
```
ggplot(
  data = normalized_trimmed,
  mapping = aes(x = Date)
) +
  geom_line(aes(y = FTSEWorldN, color = "FTSEWorldN")) +
  geom_line(aes(y = IndiaN, color = "IndiaN")) +
  geom_line(aes(y = EgyptN, color = "EgyptN")) +
  geom_line(aes(y = ChinaN, color = "ChinaN")) +
  geom_line(aes(y = BrazilN, color = "BrazilN")) +
  scale_color_manual(values = c("FTSEWorldN" = "red", "IndiaN" = "green", "EgyptN" = "blue
  labs(
    title = "Returns 20 years",
    x = "Date",
    y = "Return",
    color = "Countries"
  ) +
  theme_minimal()
```



The Graphs above shows the development for the stocks and the benchmark (normalized to 100$ invested for comparison).

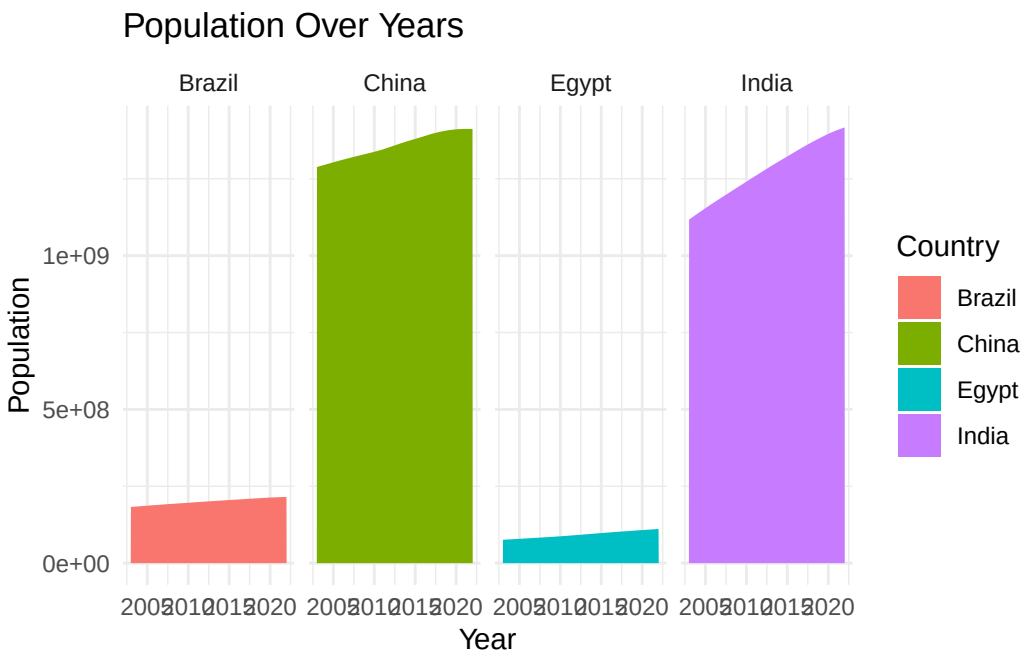
The next graph we want to show is the population development over time:

```
pop_clean_long <- population_clean %>%
  pivot_longer(cols = -Date, names_to = "Country", values_to = "Population") %>%
  mutate(Population = as.numeric(Population))

ggplot(pop_clean_long, aes(x = Date, y = Population, color = Country, group = Country)) +
  geom_line(linewidth = 1) +
  labs(title = "Population Development Over Time",
       x = "Year",
       y = "Population",
       color = "Country") +
  theme_minimal() +
  scale_y_continuous(breaks = seq(0, max(pop_clean_long$Population), by = 250000000), labe
```



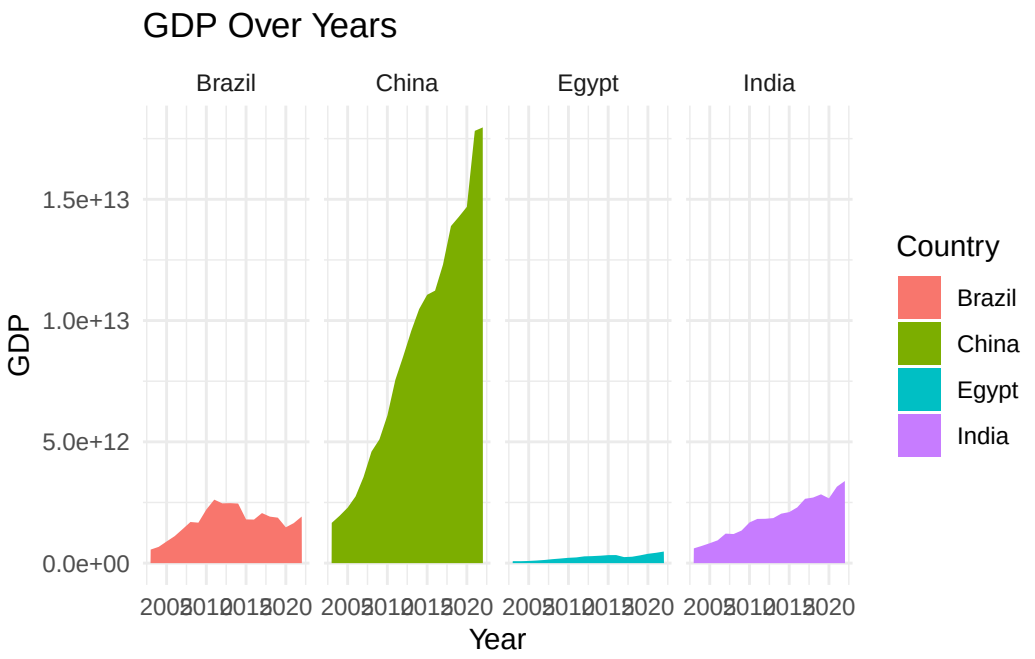
Since the population size differs quite a bit, a better visualization would be the following:

```
ggplot(population_clean_long, aes(x = Date, y = Population, fill = Country)) +
  geom_area() +
  labs(title = "Population Over Years",
       x = "Year",
       y = "Population",
       fill = "Country") +
  theme_minimal() +
  facet_grid(.~ Country, scales = "free_y")
```



In the same way we can represent the GDP:

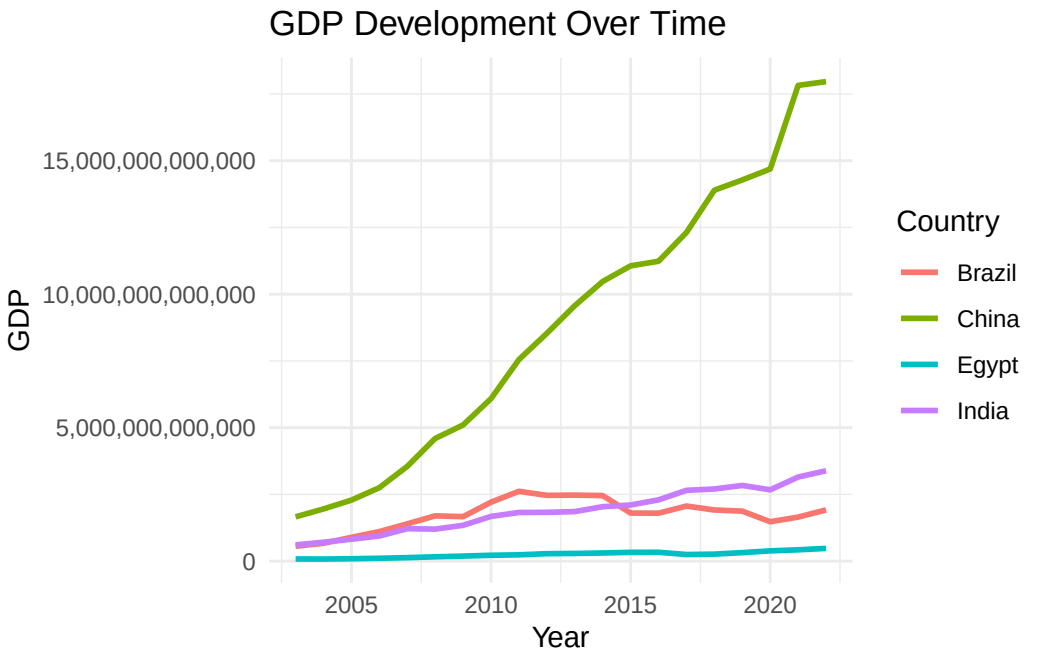
```
ggplot(gdp_clean_long, aes(x = Date, y = GDP, fill = Country)) +
  geom_area() +
  labs(title = "GDP Over Years",
       x = "Year",
       y = "GDP",
       fill = "Country") +
  theme_minimal() +
  facet_grid(. ~ Country, scales = "free_y")
```



GDP graph:

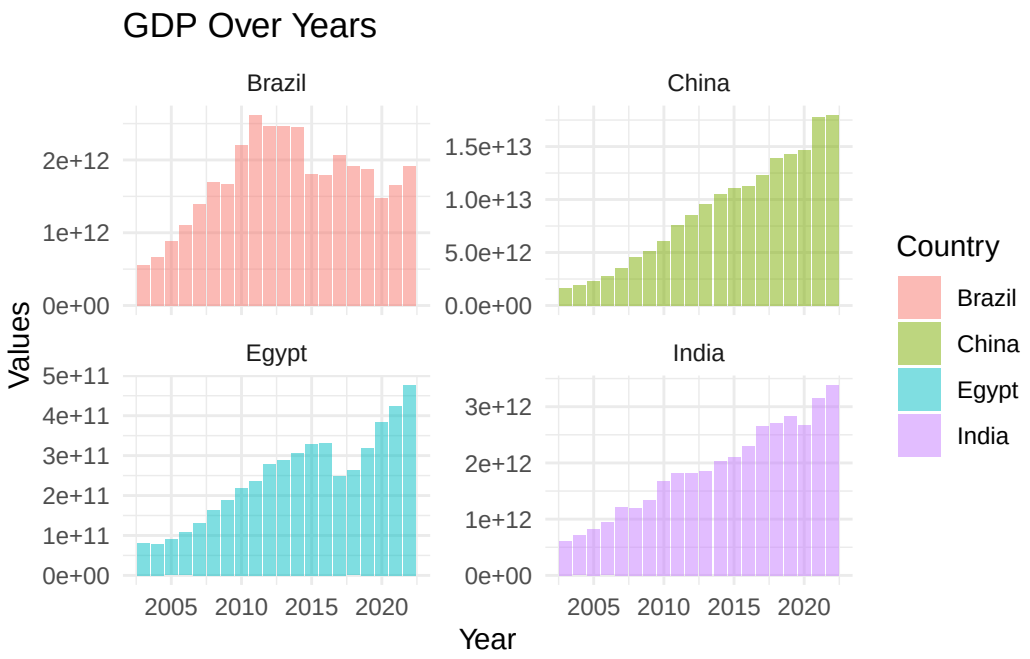
```
gdp_clean_long <- gdp_clean_long %>%
  mutate(Date = as.numeric(Date))

gdp_clean_long <- gdp_clean %>%
  pivot_longer(cols = -Date, names_to = "Country", values_to = "Population")

ggplot(gdp_clean_long, aes(x = Date, y = Population, color = Country, group = Country)) +
  geom_line(size = 1) +
  labs(title = "GDP Development Over Time",
       x = "Year",
       y = "GDP",
       color = "Country") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



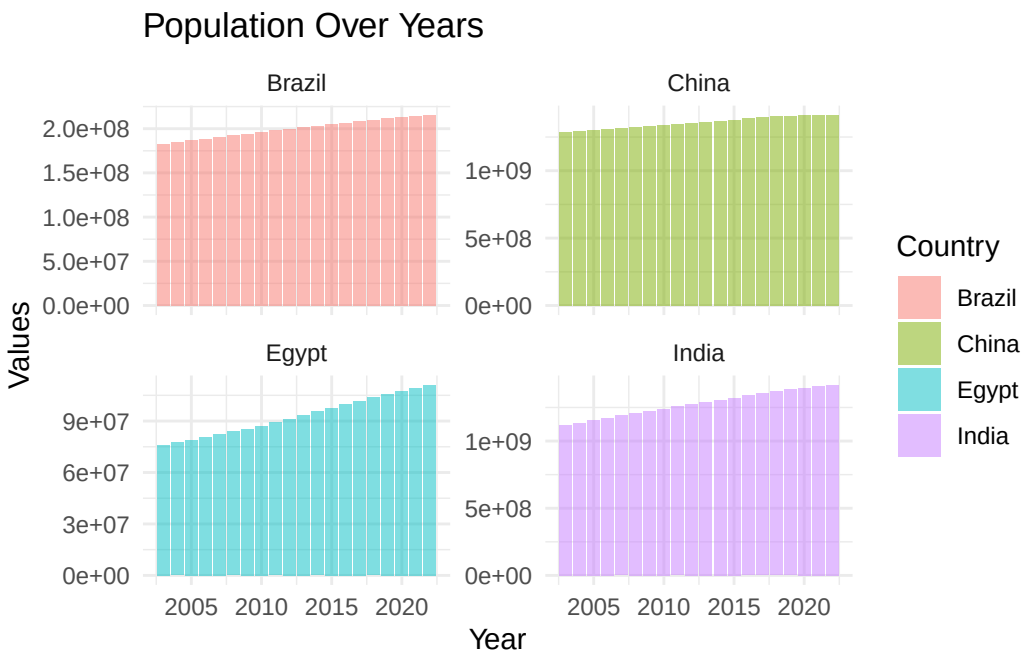
Again, we see that Egypt's GDP is much smaller compared to China. There for we can visualize it like this instead:

```
ggplot(gdp_pop, aes(x = Date)) +
  geom_bar(aes(y = GDP, fill = Country), stat = "identity", alpha = 0.5) +
  labs(title = "GDP Over Years",
       x = "Year",
       y = "Values",
       color = "Country",
       fill = "Country") +
  theme_minimal() +
  facet_wrap(~ Country, scales = "free_y")
```



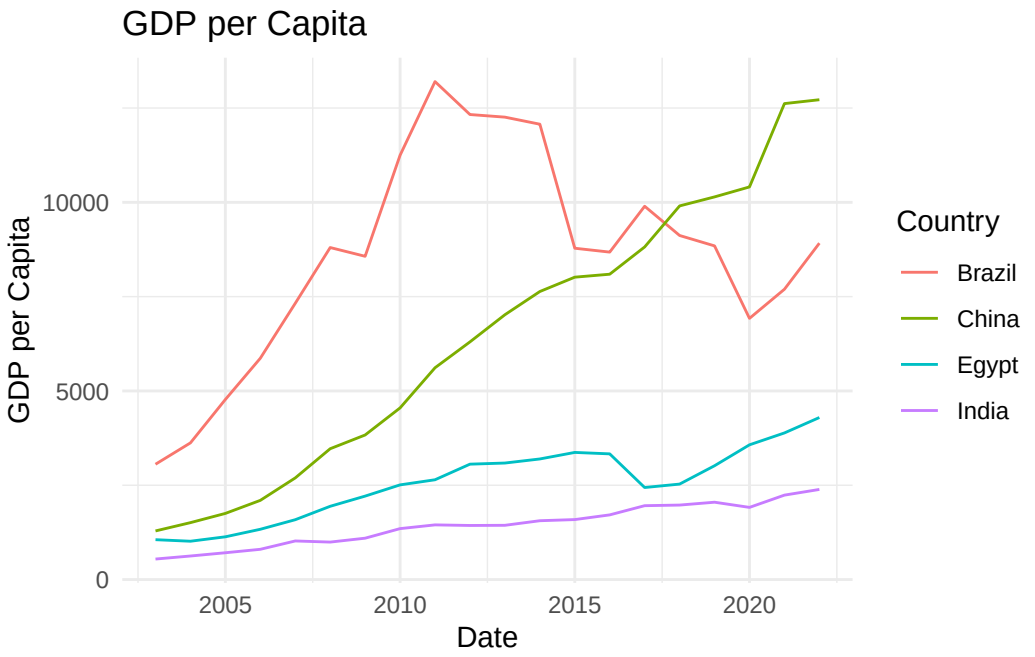
And lets do the same thing with Population:

```
ggplot(gdp_pop, aes(x = Date)) +
  geom_bar(aes(y = Population, fill = Country), stat = "identity", alpha = 0.5) +
  labs(title = "Population Over Years",
       x = "Year",
       y = "Values",
       color = "Country",
       fill = "Country") +
  theme_minimal() +
  facet_wrap(~ Country, scales = "free_y")
```



Now, lets see what GDP per capita we have for each country:

```
gdp_pop <- gdp_pop %>%
  mutate(GDP_per_Capita = GDP / Population)


ggplot(gdp_pop, aes(x = Date, y = GDP_per_Capita, color = Country)) +
  geom_line() +
  labs(title = "GDP per Capita",
       x = "Date",
       y = "GDP per Capita",
       color = "Country") +
  theme_minimal()
```



All the data we have visualized so far shows that GDP is increasing, population size is increase for each country. We can also note that the benchmark has had a positive return. We can also conclude from the graphs that indices are fluctuating a lot more compared to population and GDP.

Now I would also like to introduce some previously extracted data we can find in the exported csv in our project folder or in the data frame called final_desc.

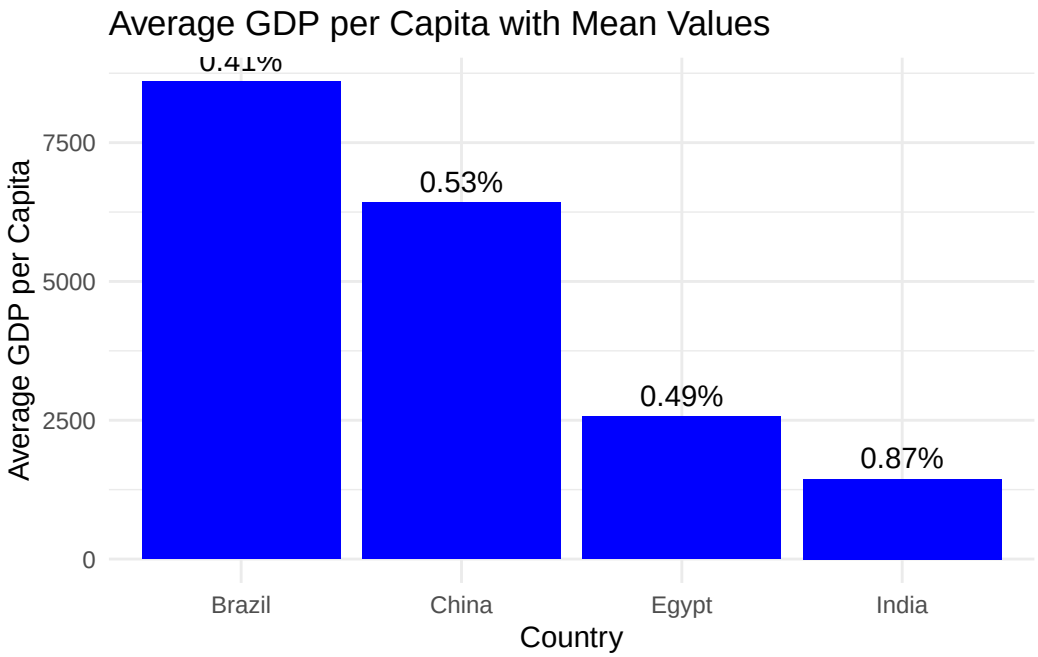
Monthly	FTSE World	Brazil	China	Egypt	India
Return (monthly)	0,47%	0,41%	0,53%	0,49%	0,87%
Return (annual)	5,74%	5,03%	6,58%	6,00%	10,98%
Risk (monthly)	5,24%	10,65%	8,02%	9,05%	8,24%
Risk (annual)	18,15%	36,88%	27,80%	31,35%	28,53%

From this overview we can quickly conclude that India by far had the best performance with a 10,98% annual return yet at a lower risk compared to both Egypt and Brazil.

FTSE World, our benchmark, clearly has the lowest overall return while also keeping risk much lower compared to the rest of the emerging markets.

To visualize the different data I produced this graph:

```r
average_gdp_per_capita <- gdp_pop %>%
  group_by(Country) %>%
  summarise(Average_GDP_Per_Capita = mean(GDP_per_Capita, na.rm = TRUE))


transposed_final_desc_data_log <- t(final_desc_data_log)
transposed_final_desc_data_log <- as.data.frame(t(final_desc_data_log))


mean_values <- transposed_final_desc_data_log %>%
  select(mean)

mean_values <- rownames_to_column(mean_values, "Country")
mean_values <- mean_values[-1, ]
mean_values$Country <- gsub("_Log", "", mean_values$Country, ignore.case = TRUE)
mean_values$Country <- gsub("EgyptN", "Egypt", mean_values$Country, ignore.case = TRUE)
merged_mean_gdp <- merge(mean_values, average_gdp_per_capita, by = "Country", all.x = TRUE

ggplot(merged_mean_gdp, aes(x = Country)) +
  geom_bar(aes(y = Average_GDP_Per_Capita), stat = "identity", fill = "blue") +
  geom_text(aes(y = Average_GDP_Per_Capita, label = paste0(round(mean * 100, 2), "%")), vj
  labs(title = "Average GDP per Capita with Mean Values",
       y = "Average GDP per Capita",
       fill = "Legend") +
  theme_minimal()
```



The data here shows bars with the GDP per capita. On top of each bar we can see the average returns for the countries FTSE index funds. We can here draw the conclusion that the funds return does not show a clear relationship. Brazil has the highest GDP per capita, yet the lowest returns over this period. Yet India with the lowest GDP per capita has had higher returns from its FTSE Fund.

Conclusion

We can here draw the conclusion that GDP per capita, representing the total output for a country over a year divided on population, does not necessarily go hand in hand with the returns of each countries FTSE index.

If we look at the data china and India have more similar sizes of population yet, India's return are much higher during our 20 year period. Egypt, China and Brazil have more similar returns.

To get back to our research question: "Is there a relationship between GDP per capita and the indices development for each country?" Based on the data we have used in this project I cannot conclude there is a a clear relationship showing. To understand these developments better we would need more extensive research with a more data and more contextualized understanding on the countries development as a whole.

The returns and risk have been summarized in this table:

Monthly	FTSE World	Brazil	China	Egypt	India
Return (monthly)	0,47%	0,41%	0,53%	0,49%	0,87%
Return (annual)	5,74%	5,03%	6,58%	6,00%	10,98%
Risk (monthly)	5,24%	10,65%	8,02%	9,05%	8,24%
Risk (annual)	18,15%	36,88%	27,80%	31,35%	28,53%

What stands out here is the almost 11% average return from India over an extended period of time. This is very impressive and as we can see well above the global average. It is also not the most volatility out of our four countries indicating its more stable i.e. a safer investment.