

# The Big Mac Index: a tool for investors

Federico J. Caballero Ferrari

2023-01-16

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Methods / Analysis</b>	<b>3</b>
2.1	Data structure . . . . .	3
2.2	Data cleaning . . . . .	6
2.3	Data exploration and visualization . . . . .	9
2.4	Modeling approach . . . . .	29
<b>3</b>	<b>Results</b>	<b>33</b>
<b>4</b>	<b>Conclusions</b>	<b>35</b>

# 1 Introduction

This is the last project of the Professional Certificate in Data Science Program, organized by Harvard University in the platform EDX. There are no restrictions about the dataset selected, except for the one already used in the MovieLens project.

Regarding the methods, at least two different algorithms must be used, with at least one being more advanced than linear or logistic regression for prediction problems.

The dataset selected contains 1,631 observations of 58 countries in the 01/04/2000 - 01/07/2022 period from 2 different datasets, uploaded by *Paul Mooney*.<sup>1</sup> The original creator of the dataset is The Economist, using external sources such as the International Monetary Fund (IMF), Reuters and McDonald's.

This dataset contains information about several countries such as local currency exchange prices, currency overvaluation and real GDP per capita. However, the most important data is the price of a Big Mac in each country, which is used as an indicator of internal consumer prices. The distribution of each attribute of the data will be shown and a data cleaning analysis will be performed in order to guarantee the most possible accurate results.

The Big Mac Index is frequently used in economics as a way to measure and compare the prices of consumer goods between two or more countries. For this purpose, “baskets” of products have been traditionally used instead; however, the fact that those baskets change from one country to another make this comparison very difficult. On the other hand, Big Macs always have the same composition in terms of ingredients and proportions, regardless of the country. The fact that these ingredients are varied and reflect food prices quite accurately (including bread, meat and vegetables) make the Big Mac index a very useful tool for economists.

At first sight, one would think that if ingredients and proportions are the same, the price of Big Macs should not change from one country to another when currency exchange rates are considered. However this is rarely the case, which means that many currencies have an actual purchasing power which is higher than the one shown by exchange prices. The Big Mac index shows, in fact, this gap between prices and real purchasing power. It can be expressed in many currencies, but in this project we will be using USD dollars. Our goal will be to use the Big Mac Index to predict when a currency is undervalued (and therefore, a good investment opportunity) in foreign exchange markets.

With that goal in mind, we will find the machine learning method that predicts those results most accurately. In order to train the different models, we will be splitting the data into two groups: train data and test data. The accuracy of each method will then be computed comparing the real test data with the predicted test data through RMSE and R squared. The machine learning methods utilized in the project are GLM, KNN, RF and CART. Finally, we will be analyzing the results of the two most accurate methods.

---

<sup>1</sup>paultimothymooney (23/10/2022). The Economist's Big Mac Index. Retrieved from <https://www.kaggle.com/datasets/paultimothymooney/the-economists-big-mac-index/download?datasetVersionNumber=32>

## 2 Methods / Analysis

The colors chosen for this report are cadet blue and steel blue (and its variations), instead of the default colors.

```
# Selected colors (to be used instead of the default ones)
colorsOfTheProject1 <- c("steelblue3")
colorsOfTheProject2 <- c("cadetblue3", "steelblue3")
```

### 2.1 Data structure

A description of the data included in each one of the 19 columns from the dataset is provided below:

- **date**: date of the observation
- **iso\_a3**: ISO country code
- **currency\_code**: international code for the local currency
- **name**: name of the country
- **local\_price**: price of a Big Mac in local currency units
- **dollar\_ex**: exchange rate (number of local currency units per dollar)
- **dollar\_price**: price of a Big Mac in USD dollars (according to the exchange rate)
- **USD\_raw**: Big Mac raw index, relative to USD dollars
- **EUR\_raw**: Big Mac raw index, relative to EUR euros
- **GBP\_raw**: Big Mac raw index, relative to GBP pounds
- **JPY\_raw**: Big Mac raw index, relative to JPY yen
- **CNY\_raw**: Big Mac raw index, relative to CNY yuan
- **GDP\_bigmac**: Gross Domestic Product per person in USD dollars, adjusted to Big Mac prices
- **adj\_price**: price of a Big Mac, adjusted to nominal GDP per person in USD dollars
- **USD\_adjusted**: Big Mac adjusted index, relative to USD dollars
- **EUR\_adjusted**: Big Mac adjusted index, relative to EUR euros
- **GBP\_adjusted**: Big Mac adjusted index, relative to GBP pounds
- **JPY\_adjusted**: Big Mac adjusted index, relative to JPY yen
- **CNY\_adjusted**: Big Mac adjusted index, relative to CNY yuan

Note: the Big Mac index variables show how much a local currency is over or undervalued against the reference currency (in this project, USD, EUR, GBP, JPY or CNY). For example, in April 2000 the exchange rate between USD dollars and ARS (Argentinian pesos) was 1 = 1. However, Big Macs were cheaper in the US (2,24 USD) than in Argentina (2,50 ARS). This means that, in terms of purchasing power, Argentinian pesos were overvalued against US dollars. For this reason, the raw Big Mac index relative to USD dollars for Argentina was 0.11607. When doing the same calculation with adjusted prices, the overvaluation shown is even higher (0.39117). In both cases, we could say that for a US investor, it would not have been profitable to invest in ARS pesos then. Instead, when index values are negative, it means that the local currency is undervalued, and hence a good investment opportunity.

The first step is to load the data. It could be downloaded **here**. In the following code, we will be presenting three different ways to download/use the data. By default, the code will only read the data from the online repository, but if the user prefers to download the data, there are the two other ways to do it.

```
# Loading the data - Different options

# 01. If the user only needs to read the data online:
big_mac_full_index <- read.csv(
  "https://raw.githubusercontent.com/fdcabfer/Big-Mac-Index/main/big-mac-full-index.csv")
```

```

# 02. If the user prefers to download it in their computer:
# url <- " https://github.com/fdcabfer/Big-Mac-Index/blob/main/big-mac-full-index.csv "
# download.file(url, destfile = " big-mac-full-index.csv ")
# data <- read_csv("big-mac-full-index.csv ")

# 03. If the data is already downloaded (in the active directory):
# data <- read_csv("big-mac-full-index.csv ")

```

In order to achieve a general understanding of the information of all the columns in the dataset, the structure of the data and the first 6 rows of the dataset will be shown:

```

# Data structure
str(big_mac_full_index)

```

```

## 'data.frame': 1631 obs. of 19 variables:
## $ date : chr "2000-04-01" "2000-04-01" "2000-04-01" "2000-04-01" ...
## $ iso_a3 : chr "ARG" "AUS" "BRA" "CAN" ...
## $ currency_code: chr "ARS" "AUD" "BRL" "CAD" ...
## $ name : chr "Argentina" "Australia" "Brazil" "Canada" ...
## $ local_price : num 2.5 2.59 2.95 2.85 5.9 ...
## $ dollar_ex : num 1 1.68 1.79 1.47 1.7 ...
## $ dollar_price : num 2.5 1.54 1.65 1.94 3.47 ...
## $ USD_raw : num 0.116 -0.312 -0.264 -0.134 0.549 ...
## $ EUR_raw : num 0.0501 -0.3525 -0.3078 -0.1857 0.4577 ...
## $ GBP_raw : num -0.167 -0.486 -0.451 -0.354 0.156 ...
## $ JPY_raw : num -0.0986 -0.4442 -0.4058 -0.301 0.2513 ...
## $ CNY_raw : num 1.091 0.289 0.378 0.622 1.903 ...
## $ GDP_bigmac : num 7803 29145 4823 26087 23873 ...
## $ adj_price : num 1.92 2.3 1.87 2.25 2.21 ...
## $ USD_adjusted : num 0.391 -0.283 -0.057 -0.077 0.682 ...
## $ EUR_adjusted : num NA NA NA NA NA NA NA NA NA ...
## $ GBP_adjusted : num -0.0663 -0.519 -0.367 -0.3805 0.1288 ...
## $ JPY_adjusted : num 0.101 -0.433 -0.254 -0.27 0.331 ...
## $ CNY_adjusted : num 0.9715 0.0156 0.3365 0.3081 1.3833 ...

```

```

# First 6 rows of data
head(big_mac_full_index)

```

```

##      date iso_a3 currency_code      name local_price dollar_ex
## 1 2000-04-01   ARG          ARS  Argentina      2.50      1.00
## 2 2000-04-01   AUS          AUD  Australia      2.59      1.68
## 3 2000-04-01   BRA          BRL   Brazil      2.95      1.79
## 4 2000-04-01   CAN          CAD   Canada      2.85      1.47
## 5 2000-04-01   CHE          CHF Switzerland      5.90      1.70
## 6 2000-04-01   CHL          CLP    Chile    1260.00    514.00
##  dollar_price USD_raw EUR_raw GBP_raw JPY_raw CNY_raw GDP_bigmac adj_price
## 1      2.500000 0.11607 0.05007 -0.16722 -0.09864 1.09091  7803.329  1.922652
## 2      1.541667 -0.31176 -0.35246 -0.48645 -0.44416 0.28939 29144.877  2.301550
## 3      1.648045 -0.26427 -0.30778 -0.45102 -0.40581 0.37836  4822.739  1.869734
## 4      1.938776 -0.13448 -0.18566 -0.35417 -0.30099 0.62152 26087.329  2.247266
## 5      3.470588 0.54937 0.45774 0.15609 0.25130 1.90267 23872.716  2.207948
## 6      2.451362 0.09436 0.02964 -0.18342 -0.11618 1.05023  4640.837  1.866505

```

##	USD_adjusted	EUR_adjusted	GBP_adjusted	JPY_adjusted	CNY_adjusted
## 1	0.39117	NA	-0.06626	0.10096	0.97153
## 2	-0.28335	NA	-0.51898	-0.43285	0.01563
## 3	-0.05696	NA	-0.36704	-0.25369	0.33645
## 4	-0.07698	NA	-0.38047	-0.26953	0.30809
## 5	0.68172	NA	0.12876	0.33090	1.38330
## 6	0.40514	NA	-0.05688	0.11201	0.99133

Which means that we have a total of 1,631 observations, with data divided into 19 columns.

## 2.2 Data cleaning

In order not to process data with missing values, it is necessary to observe if there is any NA.

```
# Find NA values (total)
is.na(big_mac_full_index) %>% sum()
```

```
## [1] 1742
```

As one can observe, there are 1,742 missing values in the dataset. Those missing values can be explained by two main factors:

- The euro was not in circulation in the year 2000, which means that there are no values for **EUR\_adjusted** in any of the observations made that year
- GDP data (**GDP\_bigmac**) is not available for some countries, which also impacts in lack of adjusted variables (**adj\_price**, **USD\_adjusted**, **EUR\_adjusted**, **GBP\_adjusted**, **JPY\_adjusted**, **CNY\_adjusted**)

In the first case, we can observe 272 missing values:

```
# Find NA values (EUR_adjusted)
is.na(big_mac_full_index$EUR_adjusted) %>% sum()
```

```
## [1] 272
```

In the case of GDP data, we can observe 245 missing values:

```
# Find NA values (GDP_bigmac)
is.na(big_mac_full_index$GDP_bigmac) %>% sum()
```

```
## [1] 245
```

As we have mentioned above it is important to note that, while there are 245 missing values only in the **GDP\_bigmac** variable, for those observations there are also missing values in the adjusted variables:

```
# Find NA values (adjusted variables, other than EUR_adjusted)
is.na(big_mac_full_index$adj_price) %>% sum()
```

```
## [1] 245
```

```
is.na(big_mac_full_index$USD_adjusted) %>% sum()
```

```
## [1] 245
```

```
is.na(big_mac_full_index$GBP_adjusted) %>% sum()
```

```
## [1] 245
```

```
is.na(big_mac_full_index$JPY_adjusted) %>% sum()
```

```
## [1] 245
```

```
is.na(big_mac_full_index$CNY_adjusted) %>% sum()
```

```
## [1] 245
```

Adding up the former missing values, we get to the 1,742 we had found originally. We will now remove those observations from our dataset.

First, we remove all the observations with missing values for the **GDP\_bigmac** variable:

```
# Remove NA values (GDP)  
data <- big_mac_full_index[!is.na(big_mac_full_index$GDP_bigmac),]
```

Then, we remove all the observations with missing values for the **EUR\_adjusted** variable:

```
# Remove NA values (EUR)  
data1 <- data[!is.na(data$EUR_adjusted),]
```

Once we have performed the data cleaning, we have the following structure:

```
# Data structure  
str(data1)
```

```
## 'data.frame': 1359 obs. of 19 variables:  
## $ date : chr "2001-04-01" "2001-04-01" "2001-04-01" "2001-04-01" ...  
## $ iso_a3 : chr "ARG" "AUS" "BRA" "CAN" ...  
## $ currency_code: chr "ARS" "AUD" "BRL" "CAD" ...  
## $ name : chr "Argentina" "Australia" "Brazil" "Canada" ...  
## $ local_price : num 2.5 3 3.6 3.33 6.3 ...  
## $ dollar_ex : num 1 1.98 2.19 1.56 1.73 ...  
## $ dollar_price : num 2.5 1.52 1.64 2.13 3.64 ...  
## $ USD_raw : num 0.1161 -0.3236 -0.2661 -0.0471 0.6257 ...  
## $ EUR_raw : num 0.1054 -0.3301 -0.2732 -0.0561 0.6102 ...  
## $ GBP_raw : num -0.121 -0.468 -0.422 -0.25 0.28 ...  
## $ JPY_raw : num 0.0544 -0.361 -0.3067 -0.0997 0.5359 ...  
## $ CNY_raw : num 1.091 0.267 0.375 0.785 2.046 ...  
## $ GDP_bigmac : num 7740 26772 4294 24277 23462 ...  
## $ adj_price : num 1.71 2.16 1.63 2.1 2.08 ...  
## $ USD_adjusted : num 0.5601 -0.2526 0.0775 0.0828 0.8644 ...  
## $ EUR_adjusted : num 0.2989 -0.3777 -0.1029 -0.0985 0.5522 ...  
## $ GBP_adjusted : num 0.0407 -0.5014 -0.2812 -0.2777 0.2437 ...  
## $ JPY_adjusted : num 0.413 -0.323 -0.024 -0.0193 0.6886 ...  
## $ CNY_adjusted : num 0.9174 -0.0814 0.3243 0.3307 1.2914 ...
```

```
# First 6 rows of data  
head(data1)
```

```

##      date iso_a3 currency_code      name local_price dollar_ex
## 29 2001-04-01   ARG          ARS   Argentina      2.50      1.00
## 30 2001-04-01   AUS          AUD   Australia      3.00      1.98
## 31 2001-04-01   BRA          BRL    Brazil      3.60      2.19
## 32 2001-04-01   CAN          CAD    Canada      3.33      1.56
## 33 2001-04-01   CHE          CHF Switzerland      6.30      1.73
## 34 2001-04-01   CHL          CLP    Chile     1260.00     601.00
##      dollar_price USD_raw EUR_raw GBP_raw JPY_raw CNY_raw GDP_bigmac
## 29      2.500000  0.11607  0.10541 -0.12148  0.05442  1.09091   7740.135
## 30      1.515152 -0.32359 -0.33005 -0.46756 -0.36096  0.26722  26772.013
## 31      1.643836 -0.26614 -0.27315 -0.42234 -0.30668  0.37484   4293.718
## 32      2.134615 -0.04705 -0.05615 -0.24988 -0.09969  0.78531  24276.895
## 33      3.641618  0.62572  0.61020  0.27969  0.53592  2.04572  23461.917
## 34      2.096506 -0.06406 -0.07300 -0.26327 -0.11576  0.75344   5038.497
##      adj_price USD_adjusted EUR_adjusted GBP_adjusted JPY_adjusted CNY_adjusted
## 29      1.708568      0.56012      0.29888      0.04071      0.41302      0.91742
## 30      2.161363     -0.25256     -0.37771     -0.50140     -0.32303     -0.08137
## 31      1.626573      0.07754     -0.10289     -0.28120     -0.02405      0.32433
## 32      2.102001      0.08277     -0.09853     -0.27772     -0.01932      0.33075
## 33      2.082612      0.86439      0.55220      0.24368      0.68860      1.29138
## 34      1.644293      0.35946      0.13183     -0.09315      0.23129      0.67081

```

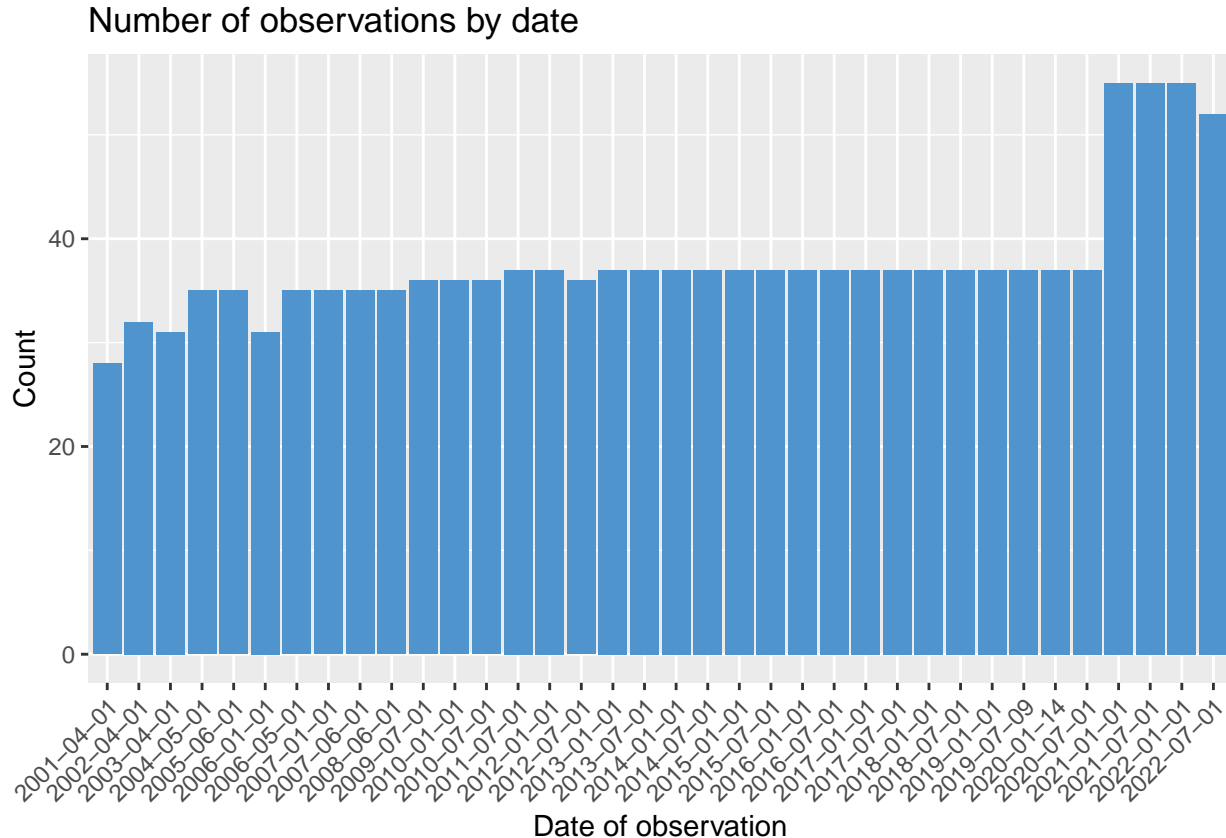
Which leaves us with a total of 1,359 observations, with data divided into 19 columns.



## 2.3 Data exploration and visualization

The first variable to explore is the date when the observations were made, shown in the **date** column:

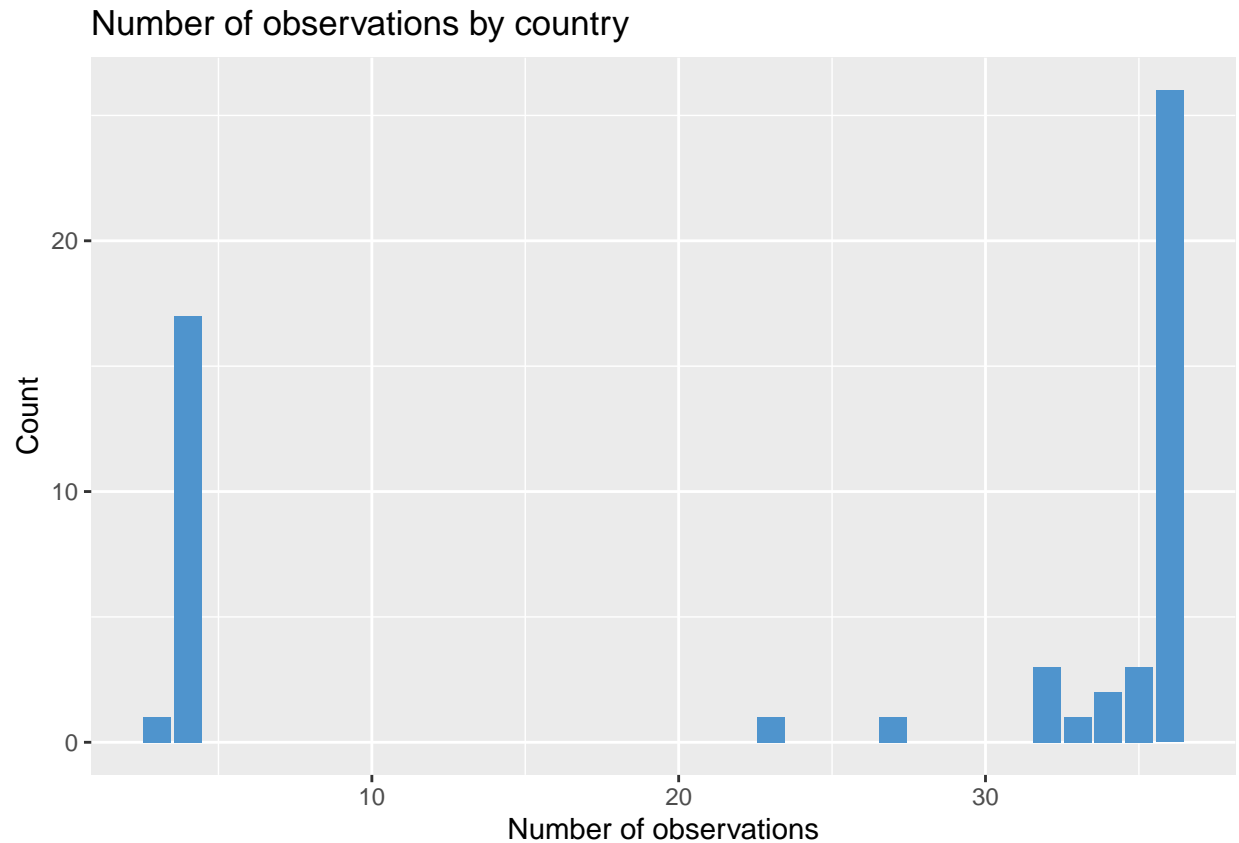
```
# Distribution of dates
d1 <- data1 %>% ggplot(aes(date)) + geom_bar(fill=colorsOfTheProject1) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1)) +
  xlab("Date of observation") + ylab("Count") + ggtitle("Number of observations by date")
d1
```



As we can see, there are more observations from recent years than from the first years in the dataset time frame, which means that we will be working with mostly updated data.

Another feature to take into account is the number of observations per country, in order to check if some countries have more relative weight in the results than others. We will show this by grouping the observations using the **iso\_a3** variable (ISO country code) as reference:

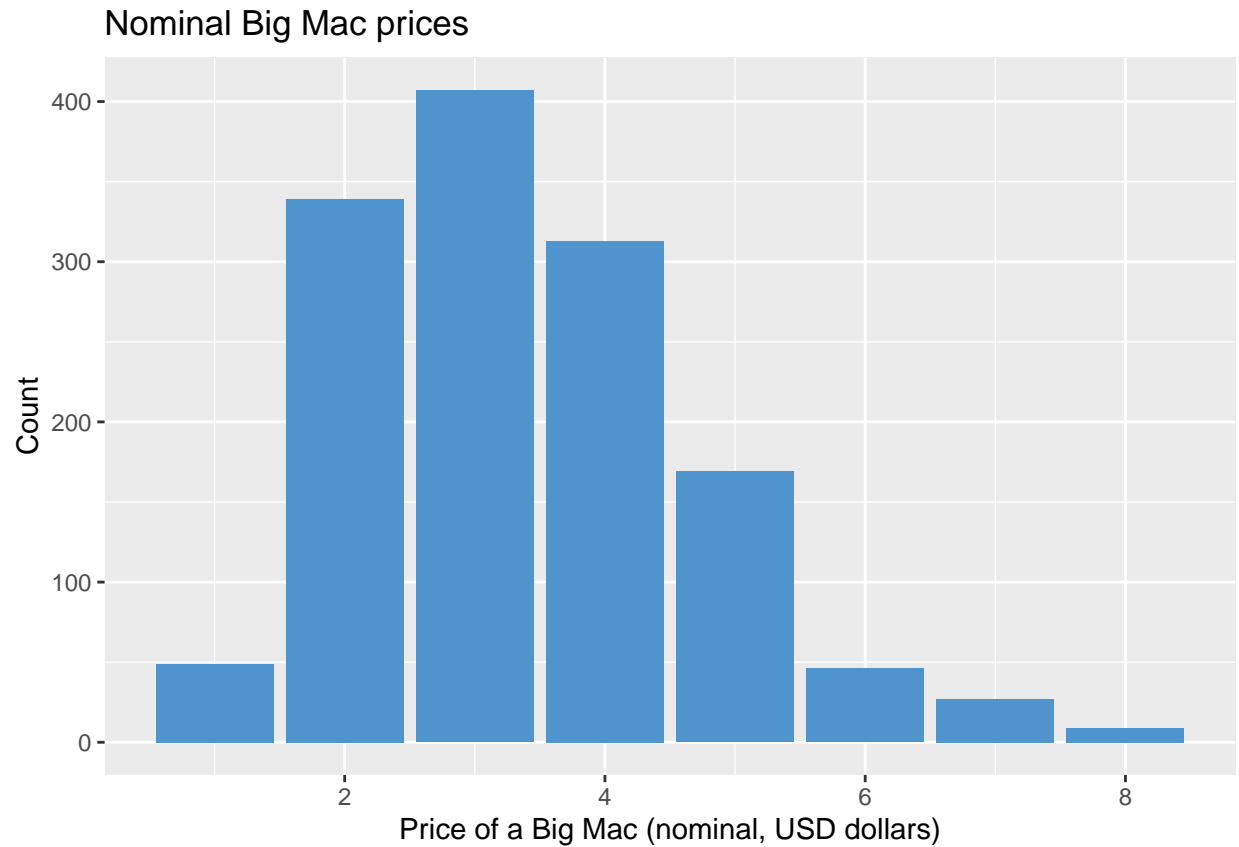
```
# Distribution of countries
data1 <- data1 %>% mutate(iso_a3 = as.factor(iso_a3))
data2 <- data1 %>% group_by(iso_a3) %>% summarise(n = n())
c1 <- data2 %>% ggplot(aes(n)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Number of observations") + ylab("Count") + ggtitle("Number of observations by country")
c1
```



As we can see, the number of observations is not always the same for every country. However, most countries have more than 20 observations for a time frame of 20 years, which means that the dataset contains consistent information across the referred period.

Now, the next variables to explore are Big Mac prices in USD dollars. In this dataset, there are two variables which represent this value. The first one is **dollar\_price**, which shows the nominal price in each country:

```
# Distribution of nominal prices
data1$dollar_price_round <- round(data1$dollar_price, digits = 0)
c2 <- data1 %>% ggplot(aes(dollar_price_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Price of a Big Mac (nominal, USD dollars)") + ylab("Count") +
  ggtitle("Nominal Big Mac prices")
c2
```



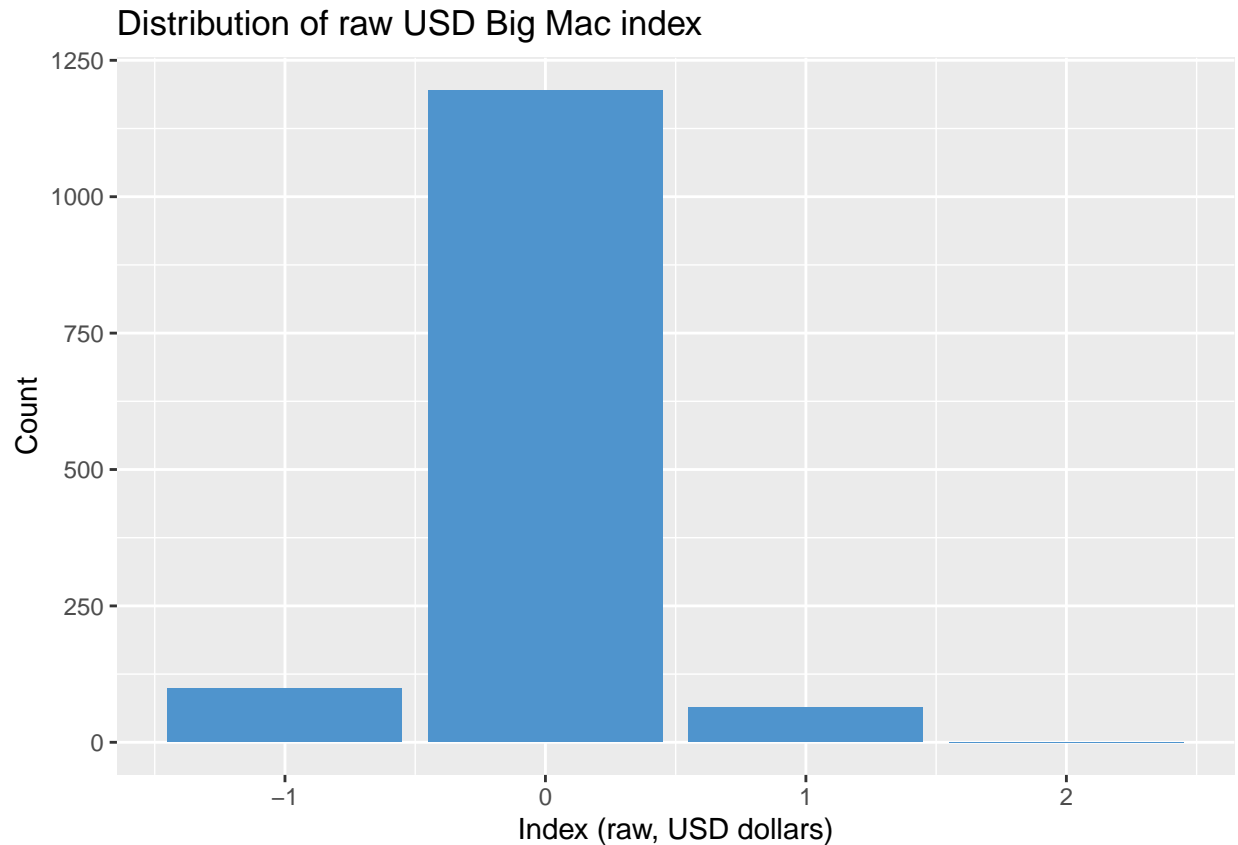
The second one is **adj\_price**, which shows the price in each country adjusted to nominal GDP per person:

```
# Distribution of adjusted prices
data1$adj_price_round <- round(data1$adj_price, digits = 0)
c3 <- data1 %>% ggplot(aes(adj_price_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Price of a Big Mac (adjusted, USD dollars)") + ylab("Count") +
  ggtitle("Adjusted Big Mac prices")
c3
```



Next, we will explore the **USD\_raw** index. It stores the first important feature:  $< 0$  if the currency is undervalued (and therefore, a good opportunity for investors who operate in USD dollars),  $0$  if it is neutral and  $> 0$  if it is overvalued. The next graph represents the **USD\_raw** distribution:

```
# Distribution of the Big Mac Index (raw, USD dollars)
data1$USD_raw_round <- round(data1$USD_raw, digits = 0)
c4 <- data1 %>% ggplot(aes(USD_raw_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (raw, USD dollars)") + ylab("Count") +
  ggtitle("Distribution of raw USD Big Mac index")
c4
```

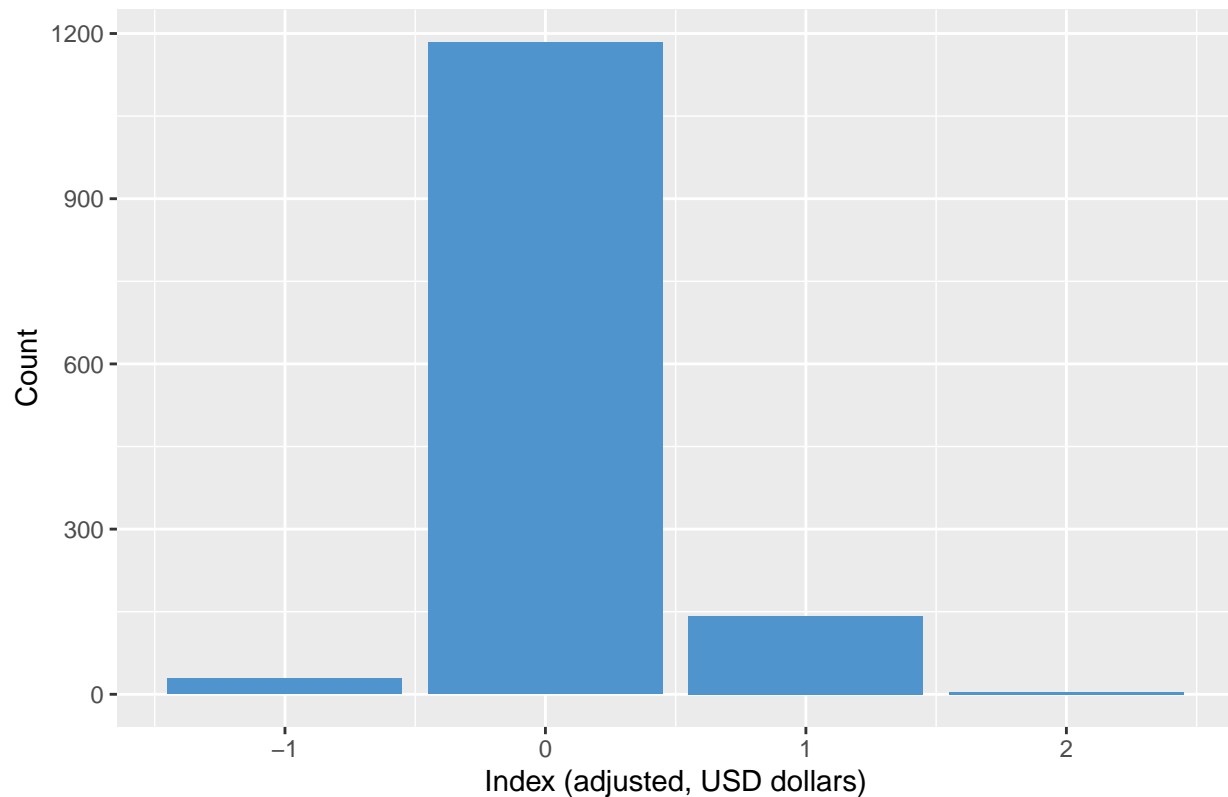


As one can observe, most values are close to 0 when rounded. Therefore, in the next section we will take a deeper analysis in order to see how many of them represent over(under)valued currencies, which means positive (negative) values.

However, there is another main feature which we will consider in this project: **USD\_adjusted**, which is the Big Mac index calculated on the adjusted price when per-capita GDP is considered, relative to USD dollars. Once again, the distribution must be read as:  $< 0$  if the currency is undervalued (and therefore, a good opportunity for investors who operate in USD dollars), 0 if it is neutral and  $> 0$  if it is overvalued. The next graph represents the **USD\_adjusted** distribution:

```
# Distribution of the Big Mac Index (adjusted, USD dollars)
data1$USD_adjusted_round <- round(data1$USD_adjusted, digits = 0)
c5 <- data1 %>% ggplot(aes(USD_adjusted_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (adjusted, USD dollars)") + ylab("Count") +
  ggtitle("Distribution of adjusted USD Big Mac index")
c5
```

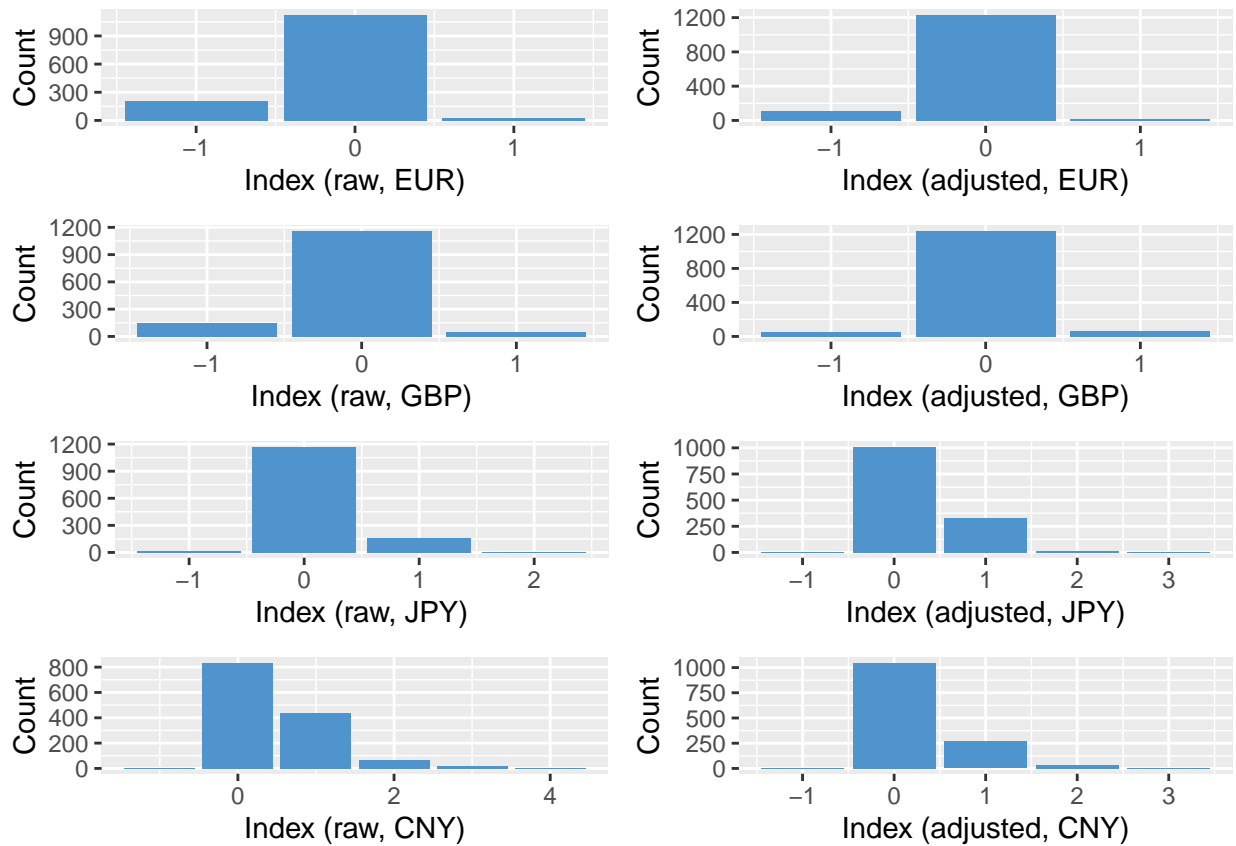
Distribution of adjusted USD Big Mac index



Similar Big Mac indexes relative to other currencies (EUR, JPY, CNY) are also presented:

```
# Distribution of the Big Mac Index (other currencies)
data1$EUR_raw_round <- round(data1$EUR_raw, digits = 0)
data1$EUR_adjusted_round <- round(data1$EUR_adjusted, digits = 0)
data1$GBP_raw_round <- round(data1$GBP_raw, digits = 0)
data1$GBP_adjusted_round <- round(data1$GBP_adjusted, digits = 0)
data1$JPY_raw_round <- round(data1$JPY_raw, digits = 0)
data1$JPY_adjusted_round <- round(data1$JPY_adjusted, digits = 0)
data1$CNY_raw_round <- round(data1$CNY_raw, digits = 0)
data1$CNY_adjusted_round <- round(data1$CNY_adjusted, digits = 0)
c6 <- data1 %>% ggplot(aes(EUR_raw_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (raw, EUR)") + ylab("Count")
c7 <- data1 %>% ggplot(aes(EUR_adjusted_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (adjusted, EUR)") + ylab("Count")
c8 <- data1 %>% ggplot(aes(GBP_raw_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (raw, GBP)") + ylab("Count")
c9 <- data1 %>% ggplot(aes(GBP_adjusted_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (adjusted, GBP)") + ylab("Count")
c10 <- data1 %>% ggplot(aes(JPY_raw_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (raw, JPY)") + ylab("Count")
c11 <- data1 %>% ggplot(aes(JPY_adjusted_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (adjusted, JPY)") + ylab("Count")
c12 <- data1 %>% ggplot(aes(CNY_raw_round)) + geom_bar(fill=colorsOfTheProject1) +
  xlab("Index (raw, CNY)") + ylab("Count")
c13 <- data1 %>% ggplot(aes(CNY_adjusted_round)) + geom_bar(fill=colorsOfTheProject1) +
```

```
xlab("Index (adjusted, CNY)") + ylab("Count")
grid.arrange(c6, c7, c8, c9, c10, c11, c12, c13, ncol=2)
```

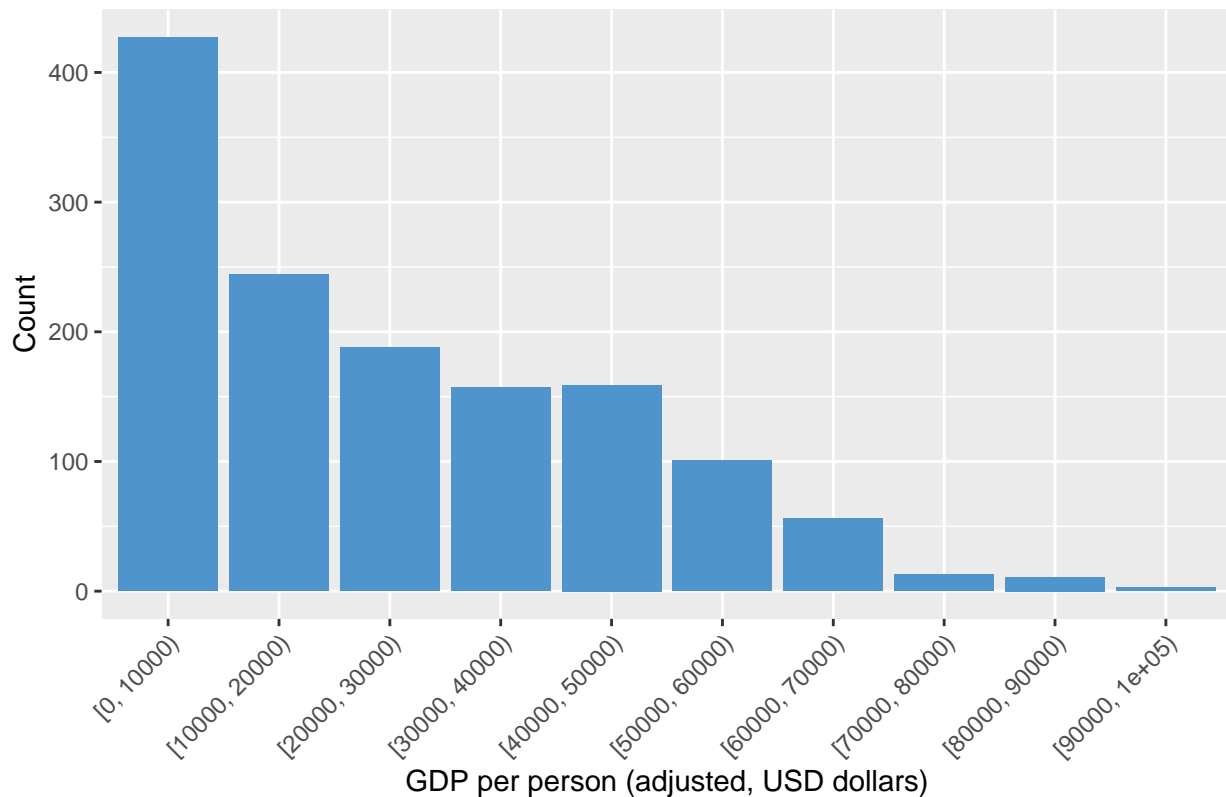


As we have seen with the variables **USD\_raw** and **USD\_adjusted**, also for these variables most values are close to 0 when rounded, so we will need further analysis.

Finally, in order to understand the variety of observations in the dataset, we will explore the distribution of the **GDP\_bigmac** feature:

```
# Distribution of GDP
data1 <- data1 %>%
  mutate(GDP_per_person = cut(GDP_bigmac, breaks = seq(0, 10^5, by = 10^4),
    labels = paste0("[", seq(0, 9*10^4, by = 10^4), ", ", seq(10^4, 10^5, by = 10^4), ")"))))
c14 <- data1 %>% ggplot(aes(GDP_per_person)) + geom_bar(fill=colorsOfTheProject1) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1)) +
  xlab("GDP per person (adjusted, USD dollars)") + ylab("Count") +
  ggtitle("Distribution of observations by GDP")
c14
```

Distribution of observations by GDP



As one can observe, most observations represent low-income countries. This can be explained by two main reasons:

- There are more low-income than high-income countries in the world
- Many of the high-income countries in the world are grouped into one (Euro Area) in this dataset

In any case, the sample has a wide range of incomes, including a significant number of middle-income countries.

There are other 4 variables which have not been shown in this section. The features **currency\_code** and **name** relate to single countries, and therefore the results we would get by grouping them would be the same that we got for **iso\_a3**. Regarding the variables **local\_price** and **dollar\_ex**, they have not been considered since their respective values are shown in local currency units and thus are not comparable to each other.

Now, for each of the features, a table and a distribution chart will be presented. These graphs will also distinguish if for each category, the local currency is undervalued or not compared to adjusted USD dollars.

In order to do this, first we will create a discrete variable (**USD\_adjusted\_discrete**) based on the **USD\_adjusted** variable. Then we will turn it into a factor (for this, we will be creating an extra variable, **USD\_adjusted\_discrete\_factor**):

```
# Create discrete USD variable, turn into factor
data1$USD_adjusted_discrete <- ifelse(data1$USD_adjusted < 0,1,0)
data1$USD_adjusted_discrete_factor <- as.factor(data1$USD_adjusted_discrete)
```

Finally, we will associate the value 1 in the factor to observations where the local currency is undervalued



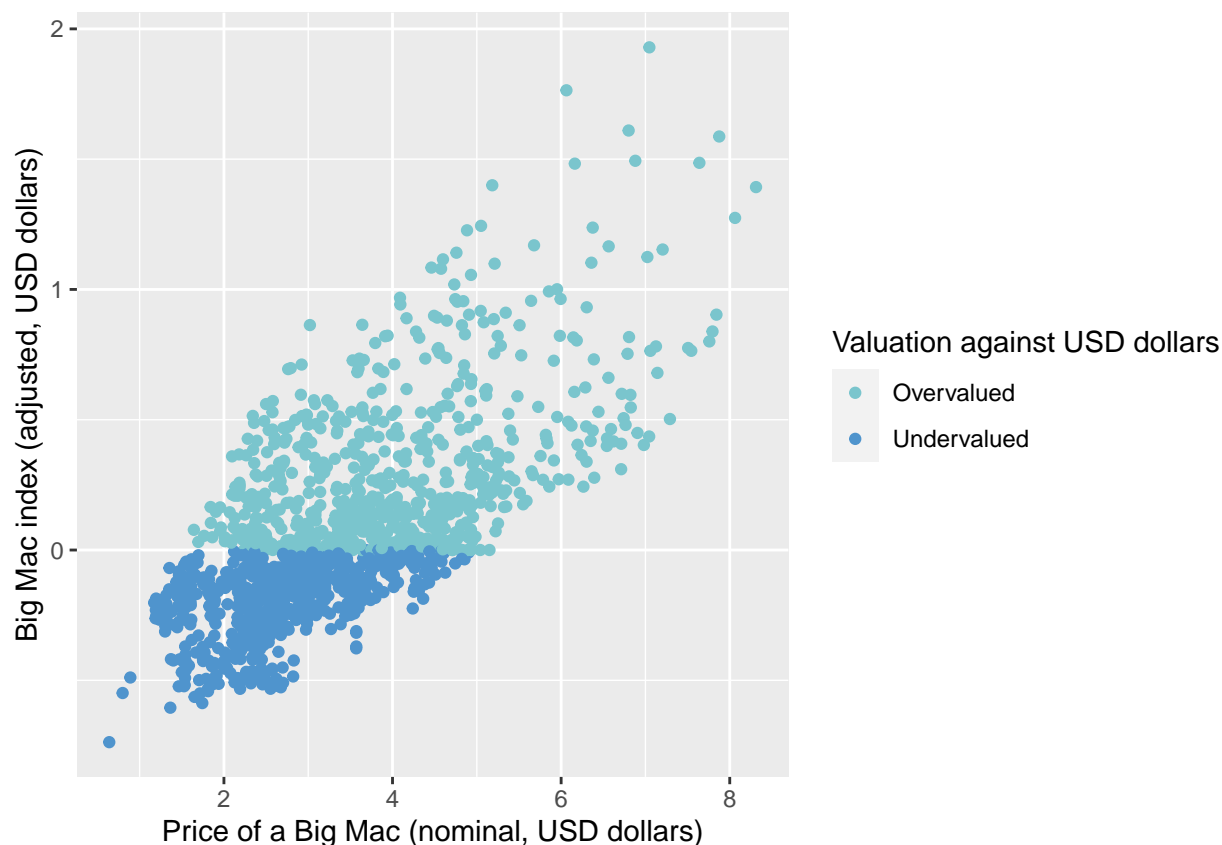
(and therefore, a good investment opportunity) and the value 0 to observations where the local currency is overvalued:

*# 1. Nominal price of a Big Mac in USD dollars*

```
data1 %>% group_by(dollar_price_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

dollar_price_round	USD_adjusted_discrete	n
1	100.00	49
2	76.99	339
3	58.72	407
4	29.39	313
5	5.92	169
6	0.00	46
7	0.00	27
8	0.00	9

```
data1 %>%
  ggplot(mapping = aes(dollar_price, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Price of a Big Mac (nominal, USD dollars)") + ylab("Big Mac index (adjusted, USD dollars)")
```



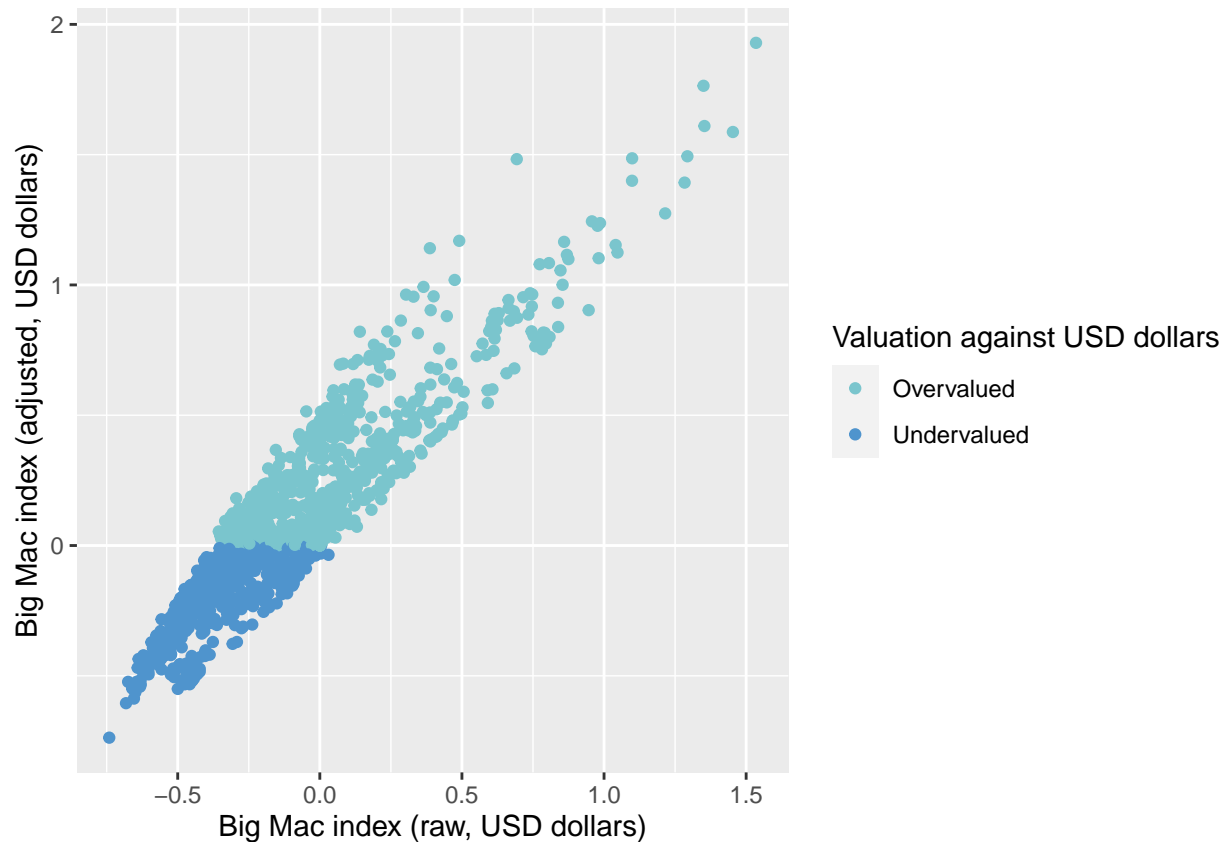
As we can observe, the higher the nominal price is, the more likely it is for the local currency to be overvalued. However, dispersion is also great, which means high variability in results.

## # 2. USD raw index

```
data1 %>% group_by(USD_raw_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

USD_raw_round	USD_adjusted_discrete	n
-1	100.00	99
0	46.19	1195
1	0.00	64
2	0.00	1

```
data1 %>%
  ggplot(mapping = aes(USD_raw, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (raw, USD dollars)") + ylab("Big Mac index (adjusted, USD dollars)")
```



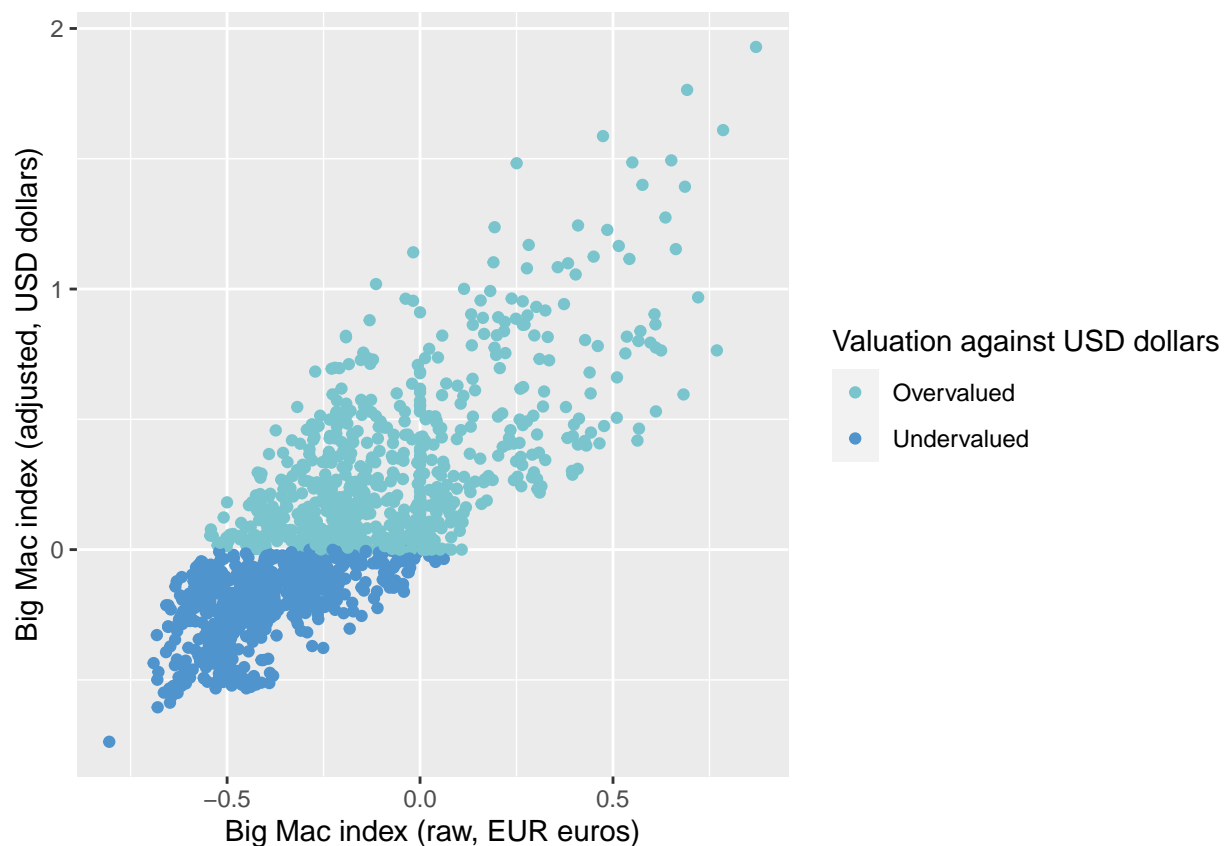
In this case, the distribution of results seems far more linear, which is logical due to the fact that we are comparing 2 variables which represent the same index (USD dollars, raw and adjusted).

### # 3. EUR raw index

```
data1 %>% group_by(EUR_raw_round) %>%
summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

EUR_raw_round	USD_adjusted_discrete	n
-1	95.61	205
0	40.41	1126
1	0.00	28

```
data1 %>%
  ggplot(mapping = aes(EUR_raw, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (raw, EUR euros)") + ylab("Big Mac index (adjusted, USD dollars)")
```



For the EUR raw index, we see positive correlation with high dispersion.

### # 4. GBP raw index

```
data1 %>% group_by(GBP_raw_round) %>%
summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

GBP_raw_round	USD_adjusted_discrete	n
-1	100.00	149
0	43.16	1163
1	0.00	47

```
data1 %>%
  ggplot(mapping = aes(GBP_raw, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (raw, GBP pounds)") + ylab("Big Mac index (adjusted, USD dollars)")
```



As we can see, a similar statement can be made about the GBP raw index.

```
# 5. JPY raw index
data1 %>% group_by(JPY_raw_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

JPY_raw_round	USD_adjusted_discrete	n
-1	100.00	18
0	53.71	1171
1	2.44	164

JPY_raw_round	USD_adjusted_discrete	n
2	0.00	6

```
data1 %>%
  ggplot(mapping = aes(JPY_raw, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (raw, JPY yen)") + ylab("Big Mac index (adjusted, USD dollars)")
```



The variability is even higher for the JPY raw index, even when the positive correlation remains. In fact, as we can see in the distribution table, 2,44 % of observations rounded to 1 show high overvaluation against JPY yen and at least some undervaluation against USD dollars.

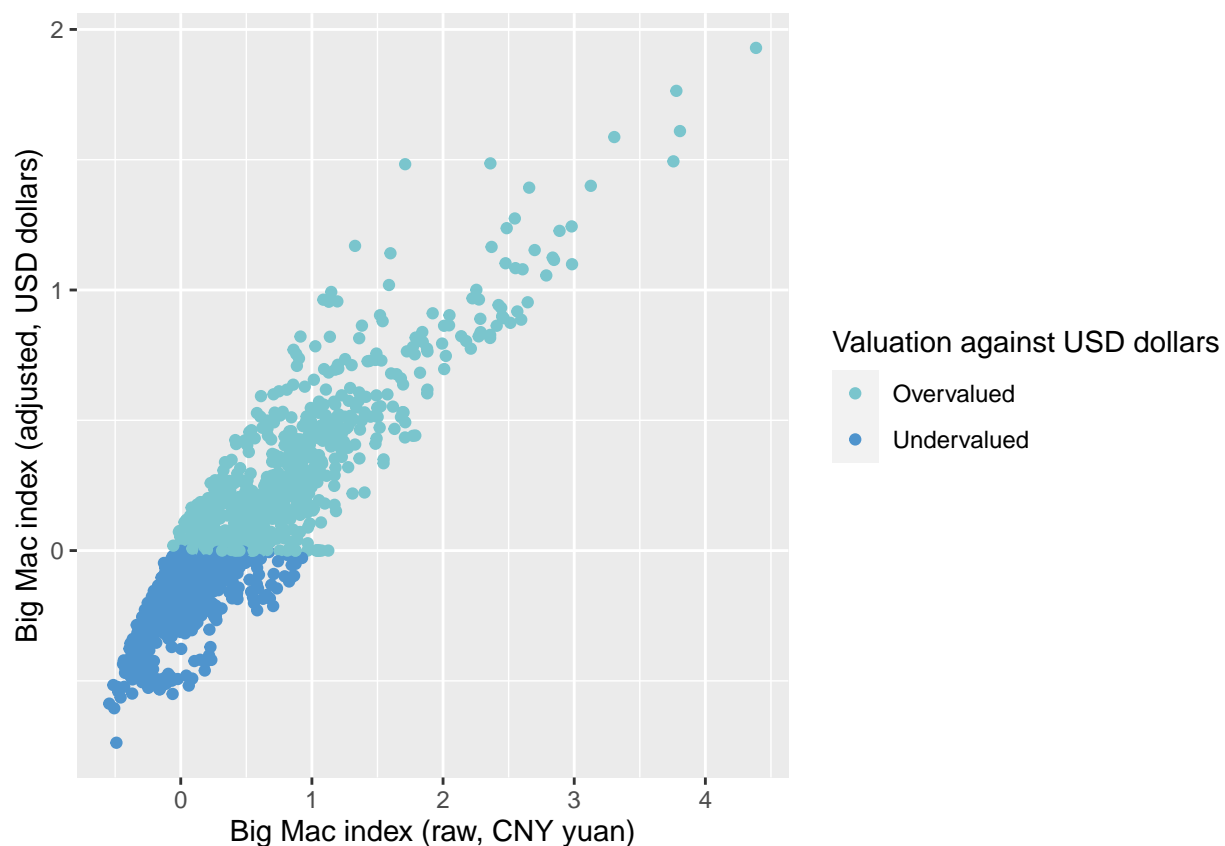
#### # 6. CNY raw index

```
data1 %>% group_by(CNY_raw_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

CNY_raw_round	USD_adjusted_discrete	n
-1	100.00	3
0	73.41	835
1	8.03	436

CNY_raw_round	USD_adjusted_discrete	n
2	0.00	64
3	0.00	17
4	0.00	4

```
data1 %>%
  ggplot(mapping = aes(CNY_raw, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (raw, CNY yuan)") + ylab("Big Mac index (adjusted, USD dollars)")
```

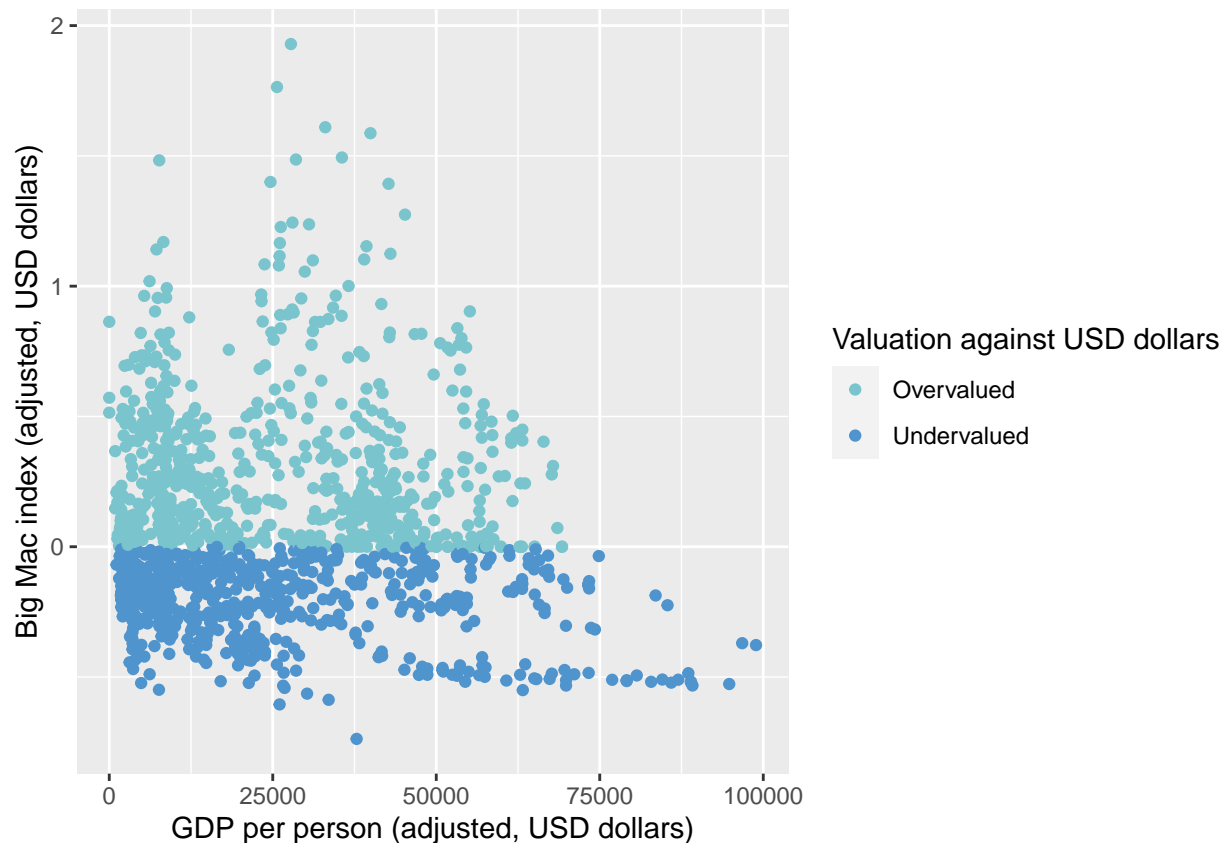


For the CNY raw index, the correlation remains positive and there is less dispersion. However, this correlation is not perfectly linear, and thus we can see that 8,03 % of observations rounded to 1 show high overvaluation against CNY yuan and at least some undervaluation against USD dollars.

```
# 7. Adjusted GDP per person
data1 %>% group_by(GDP_per_person) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

GDP_per_person	USD_adjusted_discrete	n
[0, 10000)	49.65	427
[10000, 20000)	54.10	244
[20000, 30000)	57.45	188
[30000, 40000)	33.12	157
[40000, 50000)	29.56	159
[50000, 60000)	38.61	101
[60000, 70000)	60.71	56
[70000, 80000)	100.00	13
[80000, 90000)	100.00	11
[90000, 1e+05)	100.00	3

```
data1 %>%
  ggplot(mapping = aes(GDP_bigmac, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("GDP per person (adjusted, USD dollars)") + ylab("Big Mac index (adjusted, USD dollars)")
```



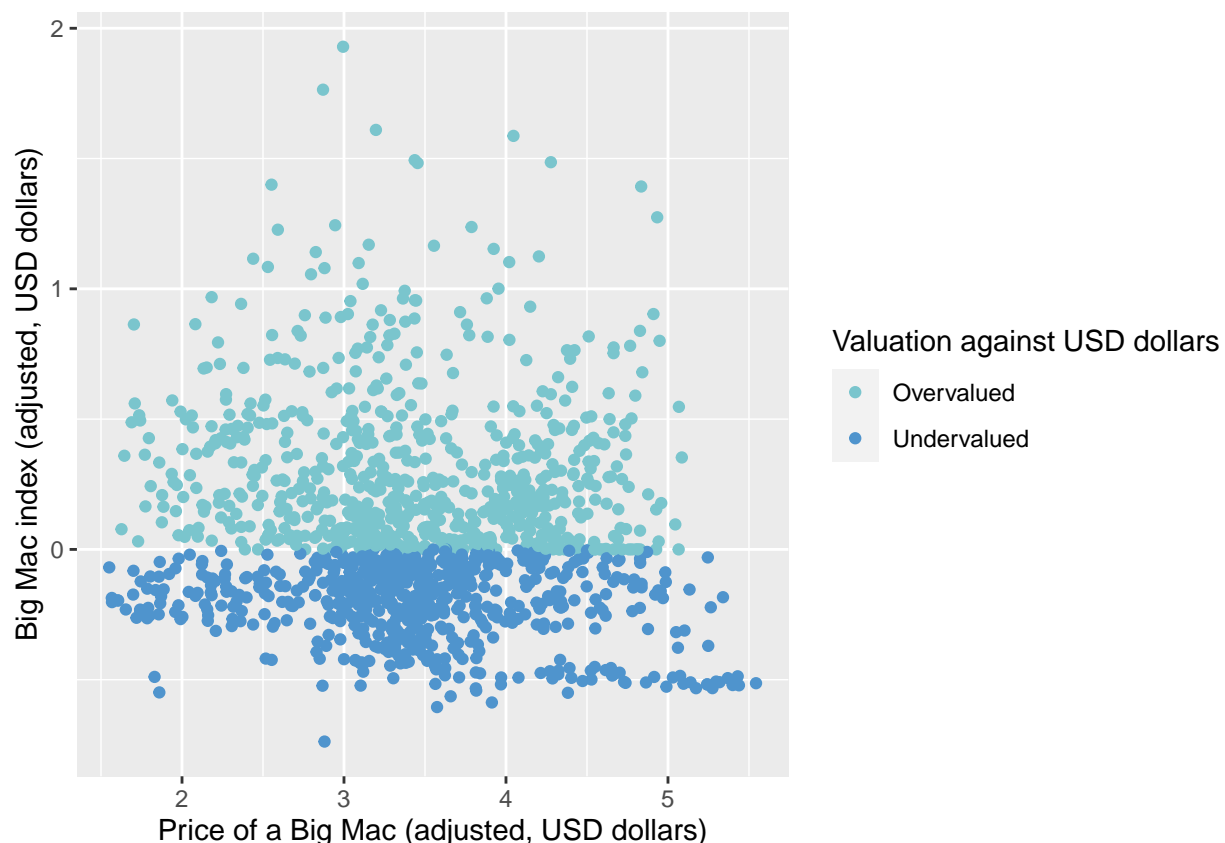
As one can observe, there is no clear correlation between adjusted GDP per person and the valuation of the local currency against USD dollars. However, in the few observations we have for the top high-income countries, there is a tendency to undervaluation. High dispersion is also observed.

```
# 8. Adjusted price of a Big Mac in USD dollars
```

```
data1 %>% group_by(adj_price_round) %>%  
summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

adj_price_round	USD_adjusted_discrete	n
2	44.05	168
3	49.57	587
4	46.45	465
5	50.00	138
6	100.00	1

```
data1 %>%  
  ggplot(mapping = aes(adj_price, USD_adjusted, color= USD_adjusted_discrete_factor)) +  
  geom_point() +  
  coord_cartesian() +  
  scale_color_manual("Valuation against USD dollars",  
                     values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +  
  xlab("Price of a Big Mac (adjusted, USD dollars)") + ylab("Big Mac index (adjusted, USD dollars)")
```



At the beginning of this section we had observed a positive correlation between nominal Big Mac prices in USD dollars and the valuation of the local currency. However, as we can see, that correlation disappears when the Big Mac prices are adjusted instead of nominal (being adjusted to the country's GDP, as we explained in section 2.1).

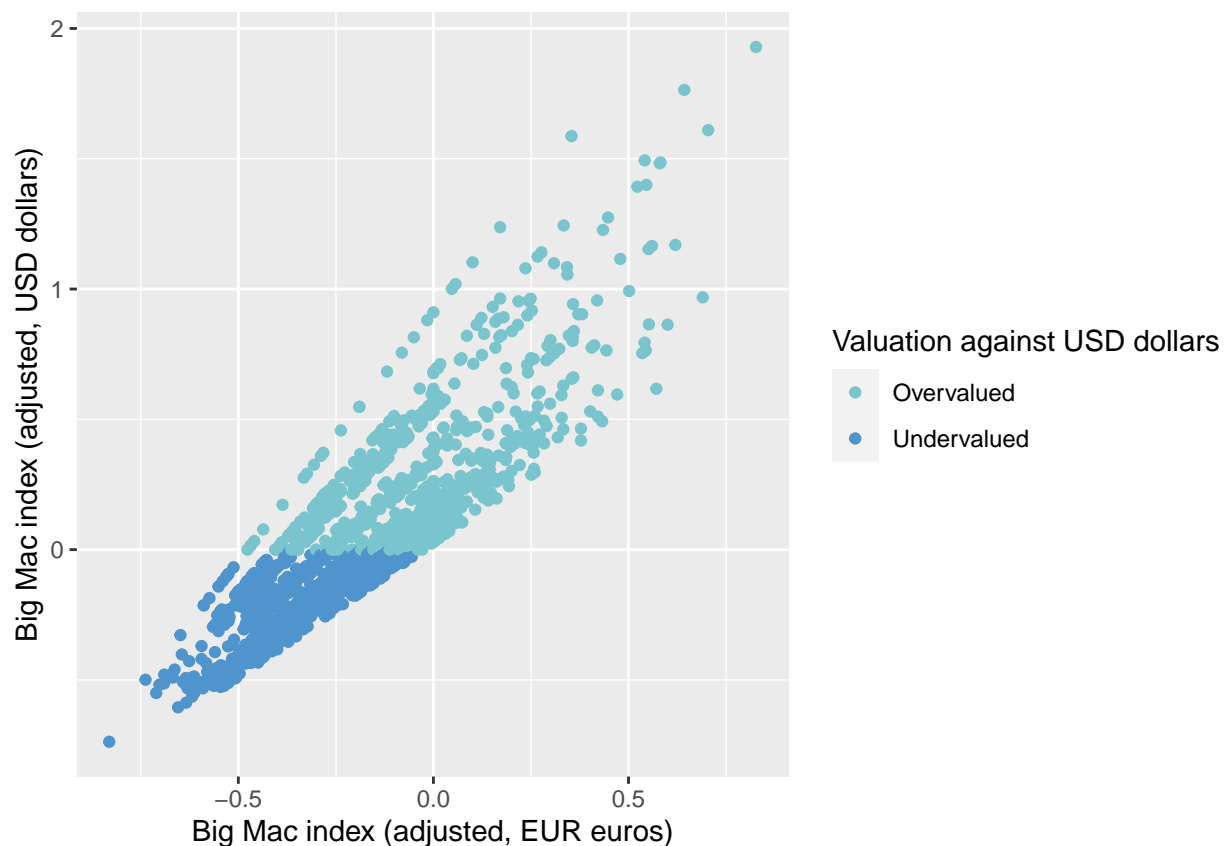


### # 9. EUR adjusted index

```
data1 %>% group_by(EUR_adjusted_round) %>%
summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

EUR_adjusted_round	USD_adjusted_discrete	n
-1	100.00	106
0	44.17	1234
1	0.00	19

```
data1 %>%
  ggplot(mapping = aes(EUR_adjusted, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian()+
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (adjusted, EUR euros)") + ylab("Big Mac index (adjusted, USD dollars)")
```



Now, if we observe the EUR adjusted index, we will see positive correlation and far less dispersion than in the raw index.

### # 10. GBP adjusted index

```
data1 %>% group_by(GBP_adjusted_round) %>%
summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

GBP_adjusted_round	USD_adjusted_discrete	n
-1	100.00	56
0	47.83	1244
1	0.00	59

```
data1 %>%
  ggplot(mapping = aes(GBP_adjusted, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian()+
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (adjusted, GBP pounds)") + ylab("Big Mac index (adjusted, USD dollars)")
```



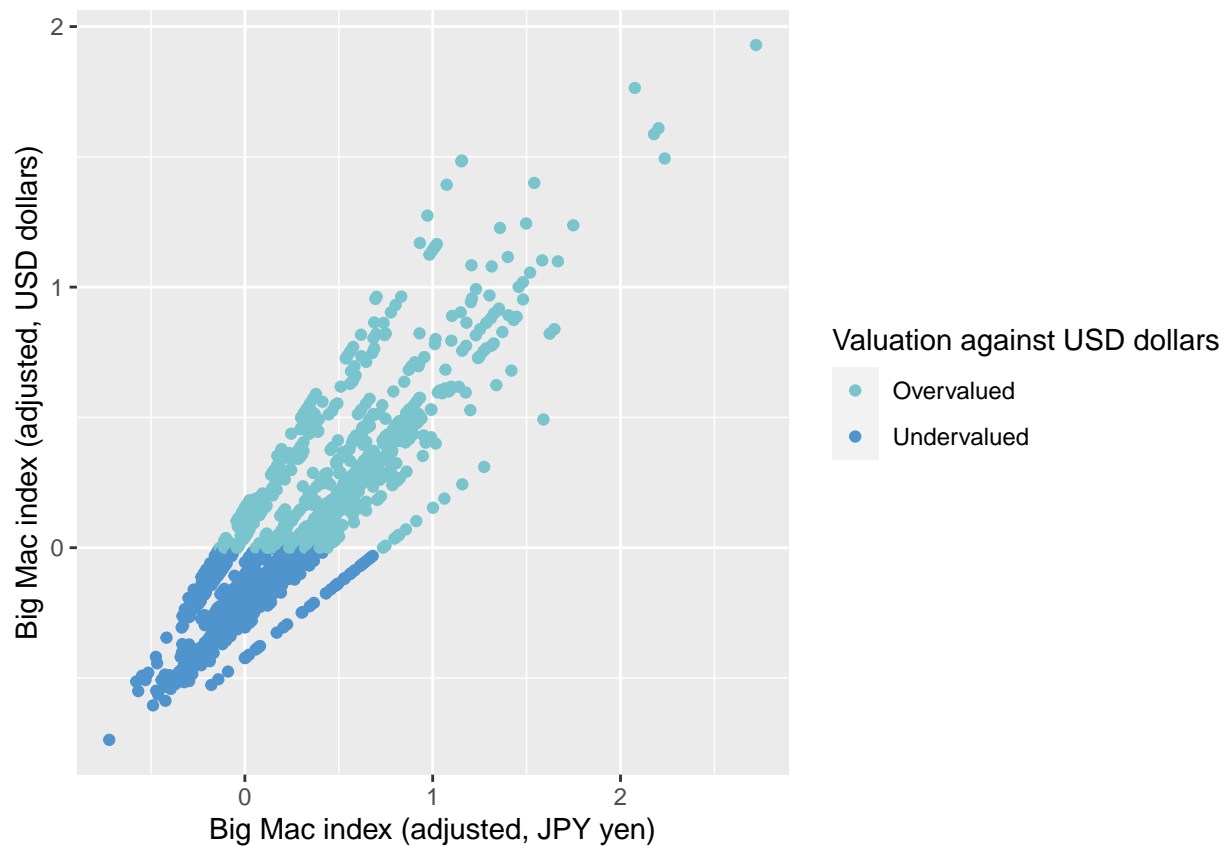
A similar statement can be made about the GBP adjusted index.

```
# 11. JPY adjusted index
data1 %>% group_by(JPY_adjusted_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

JPY_adjusted_round	USD_adjusted_discrete	n
-1	100.00	7
0	62.31	1011
1	4.27	328

JPY_adjusted_round	USD_adjusted_discrete	n
2	0.00	12
3	0.00	1

```
data1 %>%
  ggplot(mapping = aes(JPY_adjusted, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (adjusted, JPY yen)") + ylab("Big Mac index (adjusted, USD dollars)")
```



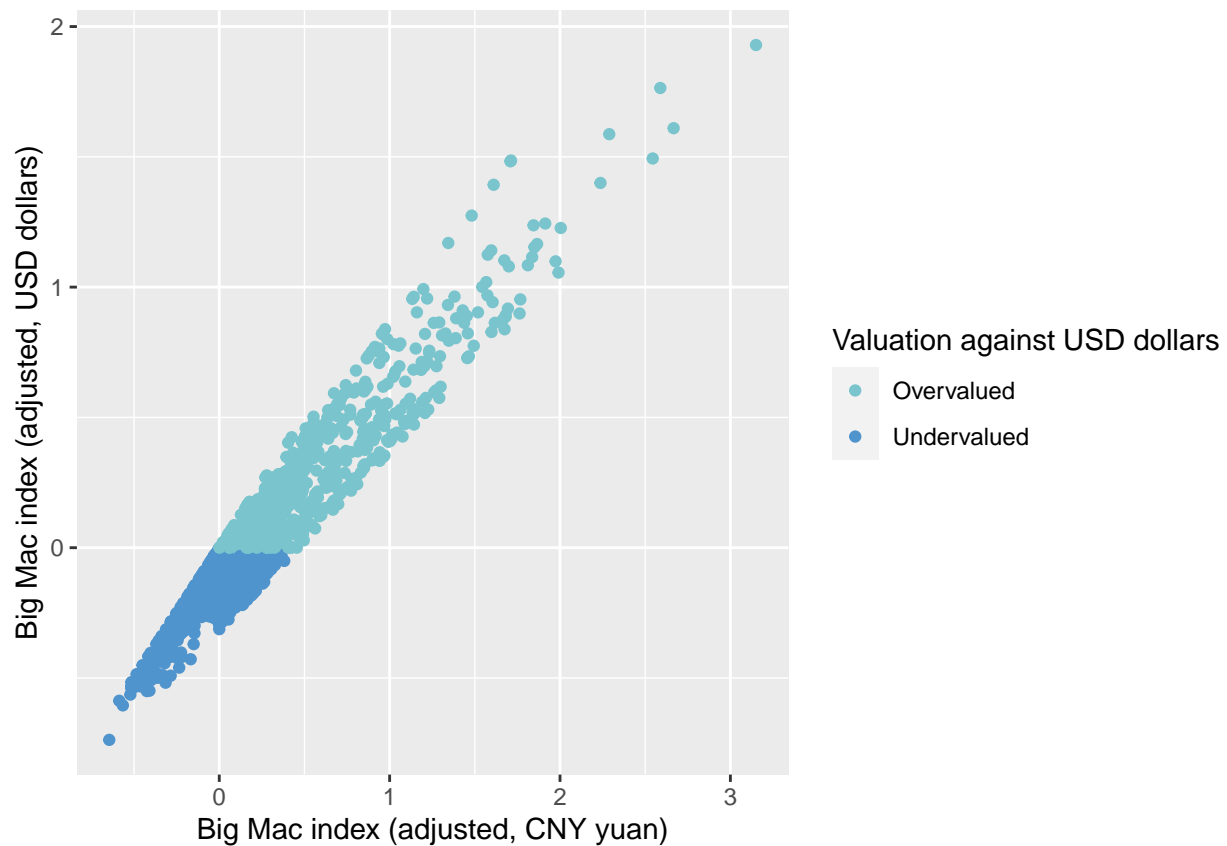
However, it is not the case for the JPY adjusted index. As we can see, there is still high variability in results even after correcting the impact of nominal GDP in prices.

```
# 12. CNY adjusted index
data1 %>% group_by(CNY_adjusted_round) %>%
  summarize(USD_adjusted_discrete = round(mean(USD_adjusted_discrete)*100,2), n=n()) %>% knitr::kable()
```

CNY_adjusted_round	USD_adjusted_discrete	n
-1	100.00	8
0	61.35	1048
1	0.00	267

CNY_adjusted_round	USD_adjusted_discrete	n
2	0.00	32
3	0.00	4

```
data1 %>%
  ggplot(mapping = aes(CNY_adjusted, USD_adjusted, color= USD_adjusted_discrete_factor)) +
  geom_point() +
  coord_cartesian() +
  scale_color_manual("Valuation against USD dollars",
                    values=colorsOfTheProject2, labels = c("Overvalued", "Undervalued")) +
  xlab("Big Mac index (adjusted, CNY yuan)") + ylab("Big Mac index (adjusted, USD dollars)")
```



In the case of the CNY adjusted index, we see much less variability than in the raw index. In fact, the distribution seems almost perfectly linear.

## 2.4 Modeling approach

For modeling purposes, some variables need to be removed for prediction:

- Variables with values that are not used for the analysis (**date**, **iso\_a3**, **currency\_code**, **name**, **local\_price**, **dollar\_ex**)
- Rounded variables (**dollar\_price\_round**, **adj\_price\_round**, **USD\_raw\_round**, **USD\_adjusted\_round**, **EUR\_raw\_round**, **EUR\_adjusted\_round**, **GBP\_raw\_round**, **GBP\_adjusted\_round**, **JPY\_raw\_round**, **JPY\_adjusted\_round**, **CNY\_raw\_round**, **CNY\_adjusted\_round**, **GDP\_per\_person**)
- Discrete variables or factors (**USD\_adjusted\_discrete**, **USD\_adjusted\_discrete\_factor**)
- Raw variables, which are perfectly correlated to the adjusted ones (**dollar\_price**, **USD\_raw**, **EUR\_raw**, **GBP\_raw**, **JPY\_raw**, **CNY\_raw**). In addition, they might present some bias due to the effect of higher nominal prices (especially for non-traded goods) in richer countries. For this reason, we will be using adjusted variables instead

The variables we will be using for our prediction models are the following: **GDP\_bigmac**, **adj\_price**, **USD\_adjusted**, **EUR\_adjusted**, **GBP\_adjusted**, **JPY\_adjusted** and **CNY\_adjusted**.

```
# Clean data for modeling
data3 <- select(data1, -date, -iso_a3, -currency_code, -name, -local_price,
                 -dollar_ex, -dollar_price_round, -adj_price_round, -USD_raw_round,
                 -USD_adjusted_round, -EUR_raw_round, -EUR_adjusted_round, -GBP_raw_round,
                 -GBP_adjusted_round, -JPY_raw_round, -JPY_adjusted_round, -CNY_raw_round,
                 -CNY_adjusted_round, -GDP_per_person, -USD_adjusted_discrete,
                 -USD_adjusted_discrete_factor, -dollar_price, -USD_raw, -EUR_raw,
                 -GBP_raw, -JPY_raw, -CNY_raw)
```

Once we have cleaned the data for prediction, we will separate the test and train datasets. The test dataset is a random sample of 30 % of the total observations, while the train dataset contains the remaining 70 %:

```
# Split data into train and test datasets
set.seed(1)
sample <- sample(c(TRUE, FALSE), nrow(data3), replace=TRUE, prob=c(0.7,0.3))
train  <- data3[sample, ]
test   <- data3[!sample, ]
```

There are multiple machine learning algorithms available. In this project 4 methods are selected. These are the models used:

- Model 1: GLM
- Model 2: KNN
- Model 3: RF
- Model 4: CART

### 2.4.1 Model 1: GLM

The *Generalized Linear Model* (GLM) generalizes linear regression by allowing the linear model to be related to the response variable and allowing the magnitude of the variance of each measurement to be a function of its predicted value. It unifies various other statistical models, including linear regression, logistic regression, and Poisson regression.

```

# Predicted results for the GLM model:
BigMac_ctrl <- trainControl(method="cv", number = 10)

BigMac_glm_trained <- train(USD_adjusted ~ .,
  train,
  method = "glm",
  trControl = BigMac_ctrl
)
BigMac_glm_predicted <- predict(BigMac_glm_trained, newdata = test )

error <- BigMac_glm_predicted - test$USD_adjusted

# Calculating the Residual Mean Squared Error (RMSE):
rmse_glm <- sqrt(mean(error^2))

# Calculating the R Squared for the GLM model:
df_glm <- data.frame(
  y_actual = test$USD_adjusted,
  y_predicted = BigMac_glm_predicted)

avr_y_actual <- mean(df_glm$y_actual)

# Total sum of squares
ss_total <- sum((df_glm$y_actual - avr_y_actual)^2)

# Regression sum of squares
ss_regression <- sum((df_glm$y_predicted - avr_y_actual)^2)

# Residual sum of squares
ss_residuals <- sum((df_glm$y_actual - df_glm$y_predicted)^2)

# R2 Score
r2_glm <- 1 - ss_residuals / ss_total

```

### 2.4.2 Model 2: KNN

The second algorithm selected is **kNN** (*k-Nearest Neighbors*) which classifies neighbors with similar characteristics and chooses the most common outcome of each of these group neighbors.

```

# Predicted results for the KNN model:
BigMac_ctrl <- trainControl(method="cv", number = 10)

BigMac_knn_trained <- train(USD_adjusted ~ .,
  train,
  method = "knn",
  trControl = BigMac_ctrl
)
BigMac_knn_predicted <- predict(BigMac_knn_trained, newdata = test )

error <- BigMac_knn_predicted - test$USD_adjusted

```

```

# Calculating the Residual Mean Squared Error (RMSE):
rmse_knn <- sqrt(mean(error^2))

# Calculating the R Squared for the KNN model:
df_knn <- data.frame(
  y_actual = test$USD_adjusted,
  y_predicted = BigMac_knn_predicted)

avr_y_actual <- mean(df_knn$y_actual)

# Total sum of squares
ss_total <- sum((df_knn$y_actual - avr_y_actual)^2)

# Regression sum of squares
ss_regression <- sum((df_knn$y_predicted - avr_y_actual)^2)

# Residual sum of squares
ss_residuals <- sum((df_knn$y_actual - df_knn$y_predicted)^2)

# R2 Score
r2_knn <- 1 - ss_residuals / ss_total

```

### 2.4.3 Model 3: RF

The third model used is *Random Forest* (**RF**). This method operates by constructing a number of decision trees during training. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

```

# Predicted results for the RF model:
BigMac_ctrl <- trainControl(method="cv", number = 10)

BigMac_rf_trained <- train(USD_adjusted ~ .,
  train,
  method = "rf",
  trControl = BigMac_ctrl
)
BigMac_rf_predicted <- predict(BigMac_rf_trained, newdata = test )

error <- BigMac_rf_predicted - test$USD_adjusted

# Calculating the Residual Mean Squared Error (RMSE):
rmse_rf <- sqrt(mean(error^2))

# Calculating the R Squared for the RF model:
df_rf <- data.frame(
  y_actual = test$USD_adjusted,
  y_predicted = BigMac_rf_predicted)

avr_y_actual <- mean(df_rf$y_actual)

```

```

# Total sum of squares
ss_total <- sum((df_rf$y_actual - avr_y_actual)^2)

# Regression sum of squares
ss_regression <- sum((df_rf$y_predicted - avr_y_actual)^2)

# Residual sum of squares
ss_residuals <- sum((df_rf$y_actual - df_rf$y_predicted)^2)

# R2 Score
r2_rf <- 1 - ss_residuals / ss_total

```

#### 2.4.4 Model 4: CART

The **CART model** (*Classification and Regression Tree*) is a variation of the decision tree algorithm, used for both regression and classification tasks:

```

# Predicted results for the CART model:
BigMac_ctrl <- trainControl(method="cv", number = 10)

BigMac_rpart_trained <- train(USD_adjusted ~ .,
                             train,
                             method = "rpart",
                             trControl = BigMac_ctrl
                             )
BigMac_rpart_predicted <- predict(BigMac_rpart_trained, newdata = test )

error <- BigMac_rpart_predicted - test$USD_adjusted

# Calculating the Residual Mean Squared Error (RMSE):
rmse_rpart <- sqrt(mean(error^2))

# Calculating the R Squared for the CART model:
df_rpart <- data.frame(
  y_actual = test$USD_adjusted,
  y_predicted = BigMac_rpart_predicted)

avr_y_actual <- mean(df_rpart$y_actual)

# Total sum of squares
ss_total <- sum((df_rpart$y_actual - avr_y_actual)^2)

# Regression sum of squares
ss_regression <- sum((df_rpart$y_predicted - avr_y_actual)^2)

# Residual sum of squares
ss_residuals <- sum((df_rpart$y_actual - df_rpart$y_predicted)^2)

# R2 Score
r2_rpart <- 1 - ss_residuals / ss_total

```



### 3 Results

We will now present how each of the previous models is able to predict results in terms of accuracy. First, we show the results based on the *Coefficient of Determination* (**R squared**), where higher values mean better accuracy:

```
# Show accuracy of models: R2
results <- tibble(method = "rf", Rsquared = r2_rf) %>%
bind_rows(tibble(method="glm", Rsquared = r2_glm)) %>%
bind_rows(tibble(method="rpart", Rsquared = r2_rpart)) %>%
bind_rows(tibble(method="knn", Rsquared = r2_knn))
results %>% knitr::kable()
```

method	Rsquared
rf	0.9824389
glm	0.9809333
rpart	0.6859379
knn	0.0323812

As we can see, the most accurate models are *Random Forest* (**RF**) and the *Generalized Linear Model* (**GLM**). For further validation, we will now see the accuracy of our models based on the *Residual Mean Squared Error* (**RMSE**), where lower values mean higher accuracy:

```
# Show accuracy of models: RMSE
results <- tibble(method = "rf", RMSE = rmse_rf) %>%
bind_rows(tibble(method="glm", RMSE = rmse_glm)) %>%
bind_rows(tibble(method="rpart", RMSE = rmse_rpart)) %>%
bind_rows(tibble(method="knn", RMSE = rmse_knn))
results %>% knitr::kable()
```

method	RMSE
rf	0.0463790
glm	0.0483262
rpart	0.1961340
knn	0.3442686

Again, the most accurate models are **RF** and **GLM**.

Finally, the importance of each variable in our 2 best models is shown. We start with **RF**:

```
# Importance of variables: Random Forest (RF)
varImp(BigMac_rf_trained)$importance %>% arrange(desc(Overall)) %>% knitr::kable()
```

	Overall
CNY_adjusted	100.0000000
JPY_adjusted	25.8258337
GBP_adjusted	17.3982104

	Overall
EUR_adjusted	9.6372199
adj_price	0.1756159
GDP_bigmac	0.0000000

And next for **GLM**:

```
# Importance of variables: Generalized Linear Model (GLM)
varImp(BigMac_glm_trained)$importance %>% arrange(desc(Overall)) %>% knitr::kable()
```

	Overall
CNY_adjusted	100.000000
adj_price	15.391908
GDP_bigmac	14.259721
EUR_adjusted	13.941048
GBP_adjusted	9.727348
JPY_adjusted	0.000000

As one can observe, in both models the most important variable is **CNY\_adjusted**. Variables such as **adj\_price** and **GDP\_bigmac** are important in the *GLM* model but not in *RF*. On the other hand, **JPY\_adjusted** is important in *RF* but not in *GLM*. Finally, both models show similar values of importance for **EUR\_adjusted** and **GBP\_adjusted**.

## 4 Conclusions

As we mentioned in the introduction, the Big Mac Index is an interesting approach to currency valuation when taking into account the purchasing power of local currencies. In this project, we have widened our scope including the valuation of some of the most important international currencies (other than the US dollar), such as the euro, the yen, the yuan and the British pound. By doing so, we have shown models in which we can predict when the currency of a country is a good opportunity for investors who operate in US dollars. Also, in our models we have used adjusted values, in order to avoid the effect of higher nominal prices (especially for non-traded goods) in richer countries.

We have used four different models, two of which have turned out to be the most accurate (*RF* and *GLM*), with similar results. Regarding the variables, both models show a strong importance of the valuation of the Chinese yuan and moderate importance of the euro and the British pound. These results can lead us to some relevant conclusions.

Firstly, once the distortion of nominal prices is discounted by using adjusted variables, we see that there is no clear correlation between a country's wealth (measured by GDP per capita) and its overvaluation regarding US dollars. Therefore, according to this study there is no evidence that there are less investment opportunities in the currencies of richer countries. In a way, this conclusion can be backed-up by the fact that these countries have traditionally been a good destination for investors, especially when interest rates are high. Thus, the popular idea that emerging markets always offer better opportunities in terms of currency undervaluation is not supported by these results.

Secondly, we have also observed that when using adjusted values, nominal prices do not play a clear role in currency valuations. In economic terms, this means that purchasing power should be approached taking more into account real prices, which involve additional factors such as wages. In investment terms, this means that opportunities might not necessarily arise when there is deflation.

In addition, we see a strong correlation between how a local currency is valued regarding US dollars and how that same currency is valued regarding Chinese yuan. This fact can tell us a lot about how strong economic relations between the US and China are. But also, it might be an indicator of how monetary policy in China has been successful when trying to achieve a stable valuation of the yuan against the dollar, not only taking into account nominal prices but also purchasing power.

Finally, we also see that the euro and British pound show moderate importance in the models. This fact is consistent with the monetary policy by the Bank of England and the European Central Bank, which usually tends to follow the steps of the US Federal Reserve, especially when it comes to interest rates. However, it is clear that there is not a strong correlation between the variables in this model, which is also proven by how investment and trade flows between Europe and the US have changed between 2001 and 2022.

In conclusion, our analysis has shown that the Big Mac Index can be a useful instrument for economic research in the field of foreign investment, but also in monetary policy. In order to deepen into the conclusions we have shown, further research should be done.