API & PCA

Marc Rainer Kullmann

2019-06-20

1 Introduction

This file serves the purpose to get familiar with two concepts. Firstly, the usage of web API's. Secondly, the concept of the Principle Component Analysis (PCA). For web API's I am going to use the World Bank Data 'webstats' package, as well as the Federal Reserve Economic Data 'fredr' package. To get used to the PCA, I choose to have a more practical application, I try to build a financial risk indicator. By collecting eight indicators from the World Bank and complement it with data from the Federal Reserve, I build the data to apply the PCA. After applying the PCA, this file continues to convert the result into a relative index (by min/max principle). Lastly, I plot the results over the time frame of eleven years (2007 - 2017) into a heat map and also plot the risk index against the mean GDP growth index.

2 Financial Risk Variables

1. **Domestic Credit To Private Sector (% Of GDP)**
   Domestic credit to private sector refers to financial resources provided to the private sector by financial corporations, such as through loans, purchases of nonequity securities, and trade credits and other accounts receivable, that establish a claim for repayment. For some countries these claims include credit to public enterprises. The financial corporations include monetary authorities and deposit money banks, as well as other financial corporations where data are available (including corporations that do not accept transferable deposits but do incur such liabilities as time and savings deposits). Examples of other financial corporations are finance and leasing companies, money lenders, insurance corporations, pension funds, and foreign exchange companies.
2. **S&P Global Equity Indices (Annual % Change)** S&P Global Equity Indices measure the U.S. dollar price change in the stock markets covered by the S&P/IFCI and S&P/Frontier BMI country indices.
3. **Foreign Direct Investment, Net (BoP, Current US$)** Foreign direct investment are the net inflows of investment to acquire a lasting management interest (10 percent or more of voting stock) in an enterprise operating in an economy other than that of the investor. It is the sum of equity capital, reinvestment of earnings, other long-term capital, and short-term capital as shown in the balance of payments. This series shows total net FDI. In BPM6, financial account balances are calculated as the change in assets minus the change in liabilities. Net FDI outflows are assets and net FDI inflows are liabilities. Data are in current U.S. dollars.
4. **Portfolio Investment, Net (BoP, Current US$)** Portfolio investment covers transactions in equity securities and debt securities. Data are in current U.S. dollars
5. **Regulatory Quality: Estimate** Regulatory Quality captures perceptions of the ability of the government to formulate and implement sound policies and regulations that permit and promote private sector development. Estimate gives the country's score on the aggregate

indicator, in units of a standard normal distribution, i.e. ranging from approximately -2.5 to 2.5.

6. **Inflation, Consumer Prices (Annual %)** Inflation as measured by the consumer price index reflects the annual percentage change in the cost to the average consumer of acquiring a basket of goods and services that may be fixed or changed at specified intervals, such as yearly. The Laspeyres formula is generally used.
7. **PPP Conversion Factor, Private Consumption (LCU Per International $)** Purchasing power parity conversion factor is the number of units of a country's currency required to buy the same amounts of goods and services in the domestic market as U.S. dollar would buy in the United States. This conversion factor is for private consumption (i.e., household final consumption expenditure). For most economies PPP figures are extrapolated from the 2011 International Comparison Program (ICP) benchmark estimates or imputed using a statistical model based on the 2011 ICP. For 47 high- and upper middle-income economies conversion factors are provided by Eurostat and the Organisation for Economic Co-operation and Development (OECD).
8. **Bank Nonperforming Loans To Total Gross Loans (%)** Bank nonperforming loans to total gross loans are the value of nonperforming loans divided by the total value of the loan portfolio (including nonperforming loans before the deduction of specific loan-loss provisions). The loan amount recorded as nonperforming should be the gross value of the loan as recorded on the balance sheet, not just the amount that is overdue.

3 Analysis

```
# Make a list of our Indicators
indicator_list <- c("FS.AST.PRVT.GD.ZS", # Domestic credit to private sector \
                                         # (% of GDP)
                    "CM.MKT.INDX.ZG",    # S&P Global Equity Indices \
                                         # (annual % change)
                    "BN.KLT.DINV.CD",    # Foreign direct investment, \
                                         # net (BoP, current US$)
                    "BN.KLT.PTXL.CD",    # Portfolio investment, \
                                         # net (BoP, current US$)
                    "RQ.EST",            # Regulatory Quality: Estimate
                    "FP.CPI.TOTL.ZG",    # Inflation, consumer prices (annual %)
                    #"PA.NUS.PPP",       # Official exchange rate \
                                         # (LCU per US$, period average)
                    "GFDD.SI.02",        # Bank nonperforming loans to total \
                                         # gross loans (%)
                    "NY.GDP.MKTP.CD")    # GDP (current US$)
indicator_name <- sort(c("DmCrd",           # Domestic credit to private sector \
                                            # (% of GDP)
                         "GlbEqIndx",       # S&P Global Equity Indices \
                                            # (annual % change)
                         "FDInet",          # Foreign direct investment, \
                                            # net (BoP, current US$)
```

```
                          "Prflinv",          # Portfolio investment, \
                                              # net (BoP, current US$)
                          "RgltryQlty",       # Regulatory Quality: Estimate
                          "Infltn",           # Inflation, consumer prices \
                                              # (annual %)
                          "PPP",              # Official exchange rate \
                                              # (LCU per US$, period average)
                          "BnkNnPfmLns"))     # Bank nonperforming loans to total \
                                              # gross loans (%)
```

```
# Collect the data into a data base:
df <- as_tibble(wb(country = "countries_only", indicator = indicator_list,
                   startdate = 2007, enddate = 2017))
raw <- df
# Put the data into a long format:
df <- df %>%
  select(-c("indicator", "iso2c", "country")) %>%
  spread(indicatorID, value) %>%
  mutate(BN.KLT.DINV.CD = BN.KLT.DINV.CD / NY.GDP.MKTP.CD) %>%
  select(-c("NY.GDP.MKTP.CD")) %>%
  rename("DmCrd" = "FS.AST.PRVT.GD.ZS",
         "GlbEqIndx" = "CM.MKT.INDX.ZG",
         "FDInet" = "BN.KLT.DINV.CD",
         "Prflinv" = "BN.KLT.PTXL.CD",
         "RgltryQlty" = "RQ.EST",
         "Infltn" = "FP.CPI.TOTL.ZG",
         #"PPP" = "PA.NUS.PPP",
         "BnkNnPfmLns" = "GFDD.SI.02")
```

Please find the PPP value commented out, as we transform it into a relative measure.

```
# Select our countries:
countries <- c("BGD", "SGP", "JPN",
               "PAK", "IND", "KOR", "VNM",
               "NZL", "AUS", "LKA",
               "CHN", "HKG", "PHL", "MYS",
               "THA")
# Select our table with countries:
df <- filter(df, iso3c %in% countries)
df %>% group_by(iso3c) %>% summarize(n())
```

```
## # A tibble: 15 x 2
##    iso3c `n()`
##    <chr> <int>
##  1 AUS      11
##  2 BGD      11
##  3 CHN      11
##  4 HKG      11
##  5 IND      11
```

```
##  6 JPN      11
##  7 KOR      11
##  8 LKA      11
##  9 MYS      11
## 10 NZL      11
## 11 PAK      11
## 12 PHL      11
## 13 SGP      11
## 14 THA      11
## 15 VNM      11
```

```
# Count missing values:
missing <- as_tibble(sapply(df, function(x) sum(is.na(x)))) %>%
  mutate(names = colnames(df))
```

```
## Warning: Calling `as_tibble()` on a vector is discouraged, because the behavior is likely to
## This warning is displayed once per session.
```

```
missing
```

```
## # A tibble: 9 x 2
##   value names
##   <int> <chr>
## 1     0 iso3c
## 2     0 date
## 3     0 FDInet
## 4     0 Prflinv
## 5     2 GlbEqIndx
## 6     0 Infltn
## 7     5 DmCrd
## 8    29 BnkNnPfmLns
## 9     0 RgltryQlty
```

As we can see, Bank nonperforming loans to total gross loans (%) seems to be problematic as we have a lot of missing values. To overcome this issue, we try to optain the NA's by collecting data from Fredr.

```
# Get Bank nonperforming loans to total gross loans (%) from FredR and WorldBank
# fredr_set_key("HERE YOUR KEY")
npl <- fredr_series_search_text(
  search_text = "non-performing loans to gross loans")
# extract the series ID's for our index by hand
fred_list <- as.factor(c("DDSI02BDA156NWDB", # Bangladesh
                         "DDSI02JPA156NWDB", # Japan
                         "DDSI02SGA156NWDB", # Singapore
                         "DDSI02PKA156NWDB", # Pakistan
                         "DDSI02IDA156NWDB", # Indonesia
                         "DDSI02KRA156NWDB", # Korea
                         "DDSI02AUA156NWDB", # Australia
                         "DDSI02CNA156NWDB", # China
```

```
                          "DDSI02HKA156NWDB", # Hong Kong
                          "DDSI02PHA156NWDB", # Philippines
                          "DDSI02MYA156NWDB", # Malaysia
                          "DDSI02THA156NWDB")) # Thailand
for (i in fred_list){
  x <- paste(i)
  assign(x, (fredr(
    series_id = i,
    observation_start = as.Date("2007-01-01"),
    frequency = "a")))
}
npl_data <- NULL
npl_data <- bind_rows(DDSI02BDA156NWDB, DDSI02JPA156NWDB, DDSI02SGA156NWDB,
                      DDSI02PKA156NWDB, DDSI02IDA156NWDB, DDSI02KRA156NWDB,
                      DDSI02AUA156NWDB, DDSI02CNA156NWDB, DDSI02HKA156NWDB,
                      DDSI02PHA156NWDB, DDSI02MYA156NWDB, DDSI02THA156NWDB)
npl_fredr <- npl_data %>%
  spread(series_id, value) %>%
  rename("BGD" = DDSI02BDA156NWDB,
         "JPN" = DDSI02JPA156NWDB,
         "SGP" = DDSI02SGA156NWDB,
         "PAK" = DDSI02PKA156NWDB,
         "IDN" = DDSI02IDA156NWDB,
         "KOR" = DDSI02KRA156NWDB,
         "AUS" = DDSI02AUA156NWDB,
         "CHN" = DDSI02CNA156NWDB,
         "HKG" = DDSI02HKA156NWDB,
         "PHL" = DDSI02PHA156NWDB,
         "MYS" = DDSI02MYA156NWDB,
         "THA" = DDSI02THA156NWDB) %>%
  gather(`BGD`, `SGP`, `JPN`, `PAK`, `IDN`, `KOR`,
         `AUS`, `CHN`, `HKG`, `PHL`, `MYS`, `THA`,
         key = "iso3c", value = BnkNnPfmLns) %>%
  mutate(date = format(date, format = "%Y"))
# Collection of old Bank nonperforming loans to total gross loans (%) from WB
npl_wb <- as_tibble(wb(country = "countries_only", indicator = "FB.AST.NPER.ZS",
                       startdate = 2007, enddate = 2017)) %>%
  rename("BnkNnPfmLns" = value) %>%
  select (-c("indicatorID", "indicator", "iso2c", "country"))
# Collection of a second Database from WB
npl_wb2 <- as_tibble(wb(country = "countries_only", indicator = "GFDD.SI.02",
                        startdate = 2007, enddate = 2017)) %>%
  rename("BnkNnPfmLns" = value) %>%
  select (-c("indicatorID", "indicator", "iso2c", "country"))
# Combining all 3 Bank nonperforming loans to total \
# gross loans (%) variables into one
npl_all <- full_join(npl_wb, npl_fredr, by = c("iso3c", "date")) %>%
```

```
  full_join(npl_wb2, by = c("iso3c", "date"))
# Replace NA's with Iterative Method.
npl_all <- npl_all %>%
  mutate(BnkNnPfmLns2 =
           ifelse(!is.na(BnkNnPfmLns.x), BnkNnPfmLns.x,
                  ifelse(!is.na(BnkNnPfmLns.y), BnkNnPfmLns.y, BnkNnPfmLns))) %>%
  filter(iso3c %in% countries) %>%
  select(-c("BnkNnPfmLns.x", "BnkNnPfmLns.y", "BnkNnPfmLns")) %>%
  rename(BnkNnPfmLns = BnkNnPfmLns2)
# Merge Bank nonperforming loans to total gross loans (%) \
# into our main table
df <- df %>% select(-c("BnkNnPfmLns")) %>%
  left_join(npl_all, by = c("iso3c", "date"))
```

```
# Changing PPP into change per year
PPP_tbl <- as_tibble(wb(country = "countries_only", indicator = "PA.NUS.PPP",
                        startdate = 2006, enddate = 2017)) %>%
  filter(iso3c %in% countries) %>%
  rename("PPP" = value) %>%
  select(-c("indicatorID", "indicator", "iso2c", "country")) %>%
  group_by(iso3c) %>%
  arrange(date) %>%
  mutate(diff = (PPP - lag(PPP, default = first(PPP))) /
           lag(PPP, default = first(PPP)))  %>%
  mutate(PPP = diff,
         date = as.numeric(date)) %>%
  select(-diff) %>%
  filter(date >= 2007)
```

As PPP in (absolute) LCU, we need to make it relative. Therefore, we take the current value subtracting it with the one year lag and divide it with the one year lag. Merging it with the main table. Following our last NA-check.

```
df <- df %>%
  mutate(date = as.numeric(date)) %>%
  full_join(PPP_tbl, by = c("iso3c", "date"))
missing <- as_tibble(sapply(df, function(x) sum(is.na(x)))) %>%
  mutate(names = colnames(df))
missing
```

```
## # A tibble: 10 x 2
##    value names
##    <int> <chr>
##  1     0 iso3c
##  2     0 date
##  3     0 FDInet
##  4     0 Prflinv
##  5     2 GlbEqIndx
```

```
##  6     0 Infltn
##  7     5 DmCrd
##  8     0 RgltryQlty
##  9    11 BnkNnPfmLns
## 10     0 PPP
```

Now we can observe, that only 11 values are missing for Bank nonperforming loans to total gross loans (%), which makes us better off.

```
# Data Wrangling and Preparing for the Weights
spread <- df %>%
  gather(key, value, -iso3c, - date) %>%
  mutate(key = paste0(key, date)) %>%
  select(-date) %>%
  spread(key, value) %>%
  mutate_all(~ifelse(is.na(.), mean(., na.rm = TRUE), .))

# Select the variables according to the min/max principle
# min: BnkNnPfmLns, Infltn, GlbEqIndx, PPP
# max: RgltryQlty, DmCrd, Prflinv, FDInet

vars_to_max <- vars_select(names(spread),
                       starts_with("DmCrd"),
                       starts_with("FDInet"),
                       starts_with("Prflinv"),
                       starts_with("RgltryQlty"))

vars_to_min <- vars_select(names(spread),
                      starts_with("BnkNnPfmLns"),
                      starts_with("Infltn"),
                      starts_with("GlbEqIndx"),
                      starts_with("PPP"))

# Create Index with min/max principle
weight_min <- spread %>% select(c(vars_to_min, "iso3c")) %>%
  mutate_at(.vars = names(vars_to_min),
            .funs = funs((. - .[which.min(.)]) /
                           (.[which.max(.)] - .[which.min(.)])))
```

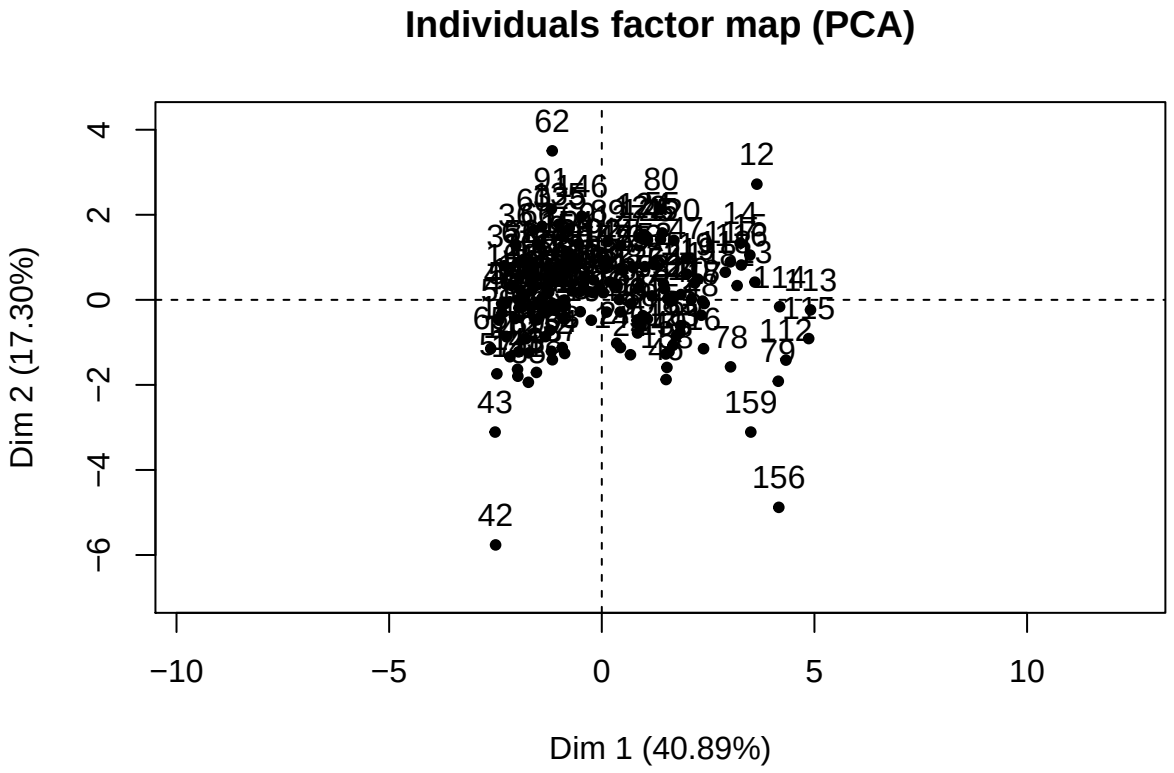
```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## please use list() instead
##
## # Before:
## funs(name = f(.)
##
## # After:
## list(name = ~f(.))
## This warning is displayed once per session.
```

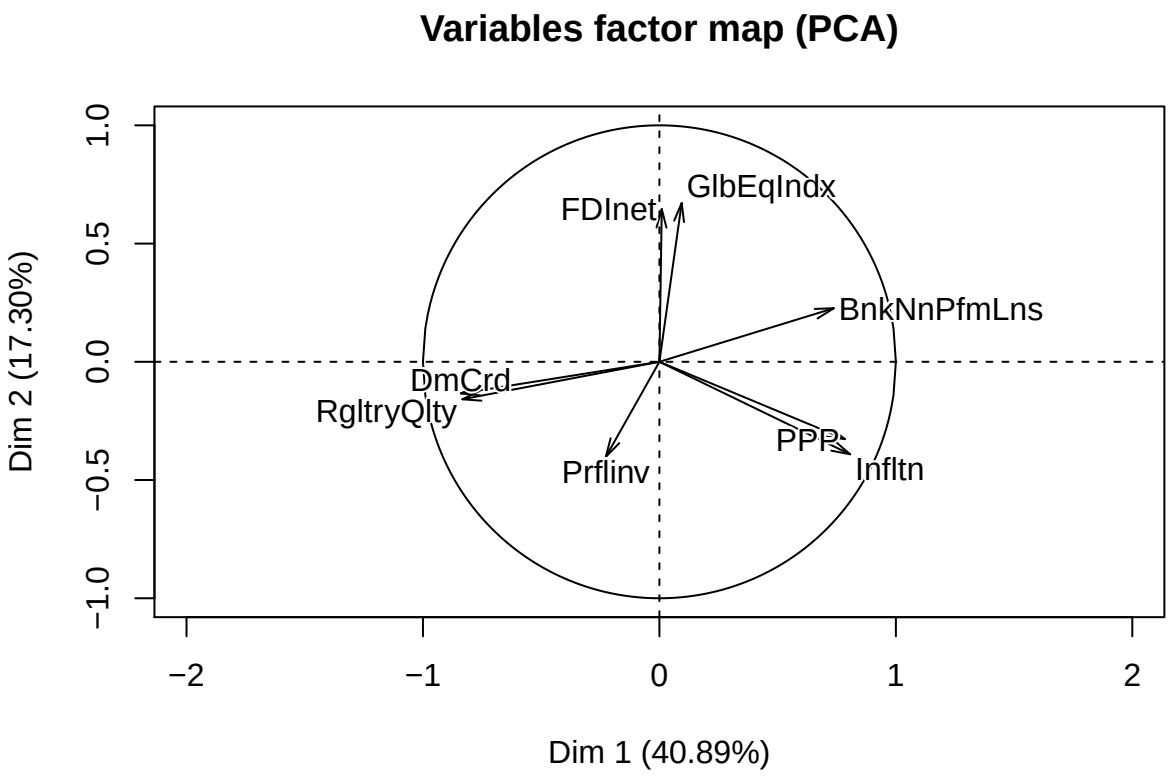
```
weight_max <- spread %>% select(c(vars_to_max, "iso3c")) %>%
  mutate_at(.vars = names(vars_to_max),
            .funs = funs((.[which.max(.)] - .) /
                         (.[which.max(.)] - .[which.min(.)])))

weighted_all <- full_join(weight_max, weight_min, by = c("iso3c"))
```

Now we have the relative weights accoring to our variables per year!

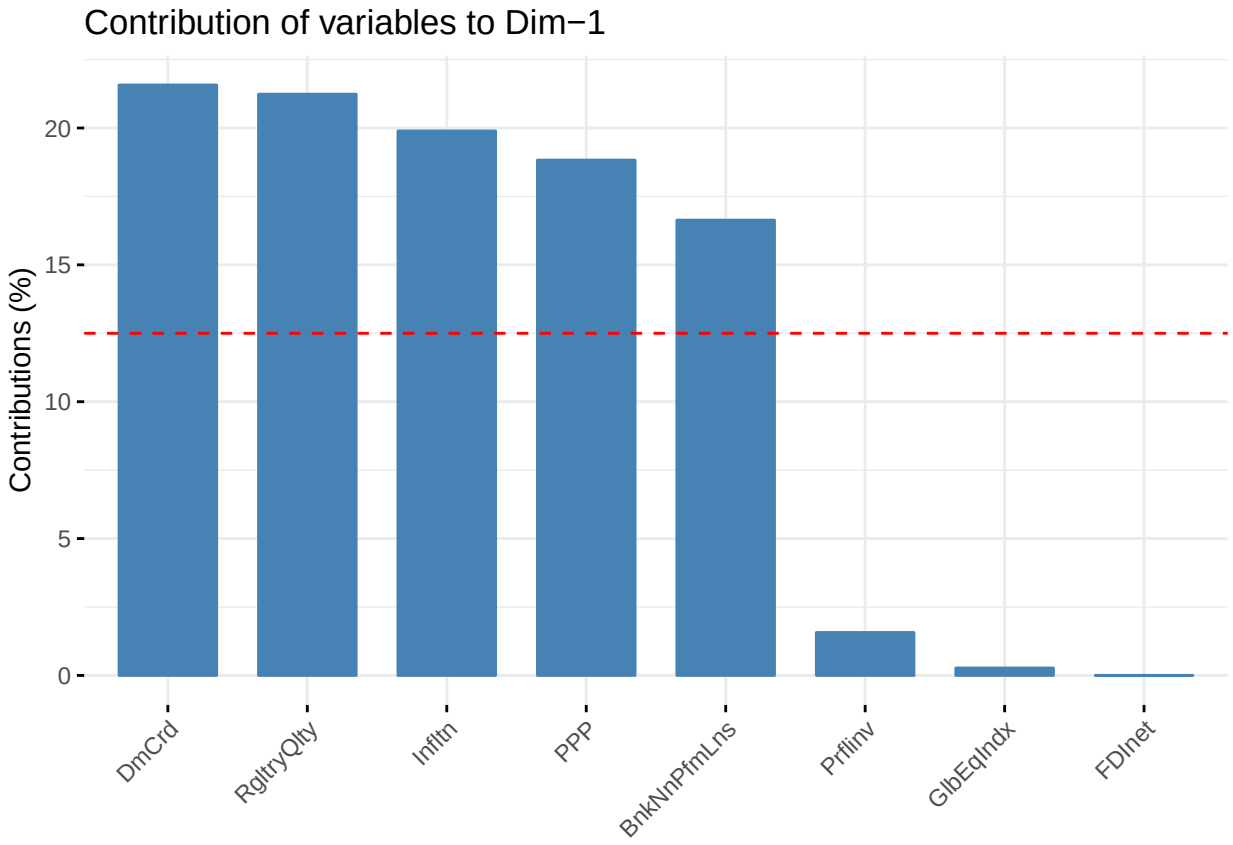
```
# Run the PCA Analysis
pca_analysis <- df %>%
  select(sort(indicator_name)) %>%
  PCA(. , graph = T)
```

```
## Warning in PCA(., graph = T): Missing values are imputed by the mean of the
## variable: you should use the imputePCA function of the missMDA package
```






```
contrib_plot <- fviz_contrib(pca_analysis, choice = "var", axes = 1)
plot(contrib_plot)
```



```r
# Create Weights according to the PCA Analysis Variables
contri <- as_tibble(pca_analysis$var$contrib) %>%
  mutate(variable = row.names(pca_analysis$var$contrib))

contri_sum <- sum(contri$Dim.1) + sum(contri$Dim.2) +
  sum(contri$Dim.3) + sum(contri$Dim.4) + sum(contri$Dim.5)

contri_weight <- mutate_at(contri, vars(-variable), funs(./sum(contri_sum)))

contri_weight_sums <- contri_weight %>%
  select(-c(variable)) %>%
  rowSums() %>% as_tibble() %>%
  t()

colnames(contri_weight_sums) <- c("BnkNnPfmLns", "DmCrd", "FDInet",
                        "GlbEqIndx", "Infltn", "PPP", "Prflinv",
                        "RgltryQlty")

contri_weight_sums <- as_tibble(contri_weight_sums) %>%
  slice(1) %>%
  mutate_all(function(x) as.numeric(as.character(x)))
contri_weight_sums
```

```
## # A tibble: 1 x 8
##   BnkNnPfmLns  DmCrd FDInet GlbEqIndx Infltn   PPP Prflinv RgltryQlty
##         <dbl>  <dbl>  <dbl>     <dbl>  <dbl> <dbl>   <dbl>      <dbl>
## 1      0.0877 0.0688  0.197     0.195 0.0870 0.113   0.196     0.0560
```

Now we have the weights from our PCA. These weights per variable will be used to create our final financial resilence indicator.

```r
# Finalize weights by multiplying Weights from PCA and Weights per Indicator
BnkNnPfmLns_years <- vars_select(names(spread), starts_with("BnkNnPfmLns"))
FDInet_years <- vars_select(names(spread), starts_with("FDInet"))
Prflinv_years <- vars_select(names(spread), starts_with("Prflinv"))
RgltryQlty_years <- vars_select(names(spread), starts_with("RgltryQlty"))
DmCrd_years <- vars_select(names(spread), starts_with("DmCrd"))
Infltn_years <- vars_select(names(spread), starts_with("Infltn"))
GlbEqIndx_years <- vars_select(names(spread), starts_with("GlbEqIndx"))
PPP_years <- vars_select(names(spread), starts_with("PPP"))

weighted_all <- weighted_all %>%
  mutate_at(.vars = BnkNnPfmLns_years,
            .funs = funs( (.) * contri_weight_sums$BnkNnPfmLns)) %>%
  mutate_at(.vars = FDInet_years,
            .funs = funs( (.) * contri_weight_sums$FDInet)) %>%
  mutate_at(.vars = Prflinv_years,
            .funs = funs( (.) * contri_weight_sums$Prflinv)) %>%
```

```
  mutate_at(.vars = RgltryQlty_years,
            .funs = funs( (.) * contri_weight_sums$RgltryQlty))%>%
  mutate_at(.vars = DmCrd_years,
            .funs = funs( (.) * contri_weight_sums$DmCrd)) %>%
  mutate_at(.vars = Infltn_years,
            .funs = funs( (.) * contri_weight_sums$Infltn))%>%
  mutate_at(.vars = GlbEqIndx_years,
            .funs = funs( (.) * contri_weight_sums$GlbEqIndx)) %>%
  mutate_at(.vars = PPP_years,
            .funs = funs( (.) * contri_weight_sums$PPP))
```

As we have our final weights now, we can create the yearly index

```
# Create yearly Variable
years <- as.character(c(2007:2017))

b <- as.list(NULL)
for (i in years){
  b[[i]] <- as_tibble(
    by_row(.d = select(
      weighted_all,
      vars_select(names(weighted_all),
                  ends_with(i))),
      ..f = sum, .to = i, .collate = "cols"))
}

# Merge final resilence indicator
variable <- as_tibble(weighted_all$iso3c)%>%
  mutate(v2007 = b$`2007`$`2007`,
         v2008 = b$`2008`$`2008`,
         v2009 = b$`2009`$`2009`,
         v2010 = b$`2010`$`2010`,
         v2011 = b$`2011`$`2011`,
         v2012 = b$`2012`$`2012`,
         v2013 = b$`2013`$`2013`,
         v2014 = b$`2014`$`2014`,
         v2015 = b$`2015`$`2015`,
         v2016 = b$`2016`$`2016`,
         v2017 = b$`2017`$`2017`
         )
```

```
index <- gather(variable, "v2007", "v2008",
                "v2009", "v2010", "v2011",
                "v2012", "v2013", "v2014",
                "v2015", "v2016", "v2017",
                key = year,value = Risk)
index <- separate(index, year, c("time", "year"), sep = 1) %>%
  select(-time)
```

```
index <- mutate(index, year = as.numeric(year)) %>%
  rename(iso3c = value)
```

```
# Download GDP Growth Rate from WorldBank
GDP_Growth <- as_tibble(
  wb(country = "countries_only", indicator = "NY.GDP.MKTP.KD.ZG",
     startdate = 2007, enddate = 2017)
  ) %>%
  filter(iso3c %in% countries) %>%
  rename("GDP Growth Rate" = value,
         "year" = date) %>%
  select (-c("indicatorID", "indicator", "iso2c", "country")) %>%
  mutate(year = as.numeric(year))
# Generate GDP Growth Rate Index
GDP_Growth_wide <- spread(GDP_Growth, year, `GDP Growth Rate`)
GDP_Growth_wide_names <- names(GDP_Growth_wide)
GDP_Growth_wide_names <-  GDP_Growth_wide_names[-1]
index_growth <- GDP_Growth_wide %>%
  mutate_at(.vars=GDP_Growth_wide_names,
            .funs=funs(( . - .[which.min(.)]) /
                         (.[which.max(.)] - .[which.min(.)]))) %>%
  gather(`2007`, `2008`, `2009`, `2010`, `2011`, `2012`,
         `2013`, `2014`, `2015`, `2016`, `2017`,
         key = year, value = g_index) %>%
  mutate(year = as.numeric(year))
# Join GDP Growth Rate Index Together
index <- index %>% full_join(index_growth, by = c("iso3c", "year"))
index <- index %>% mutate(
  mean_Risk = mean(Risk),
  mean_GDPindex = mean(g_index))
# Generating means for vertial and horizontal line
mean_Risk <- mean(index$Risk)
mean_GDPindex <- mean(index$g_index)
```

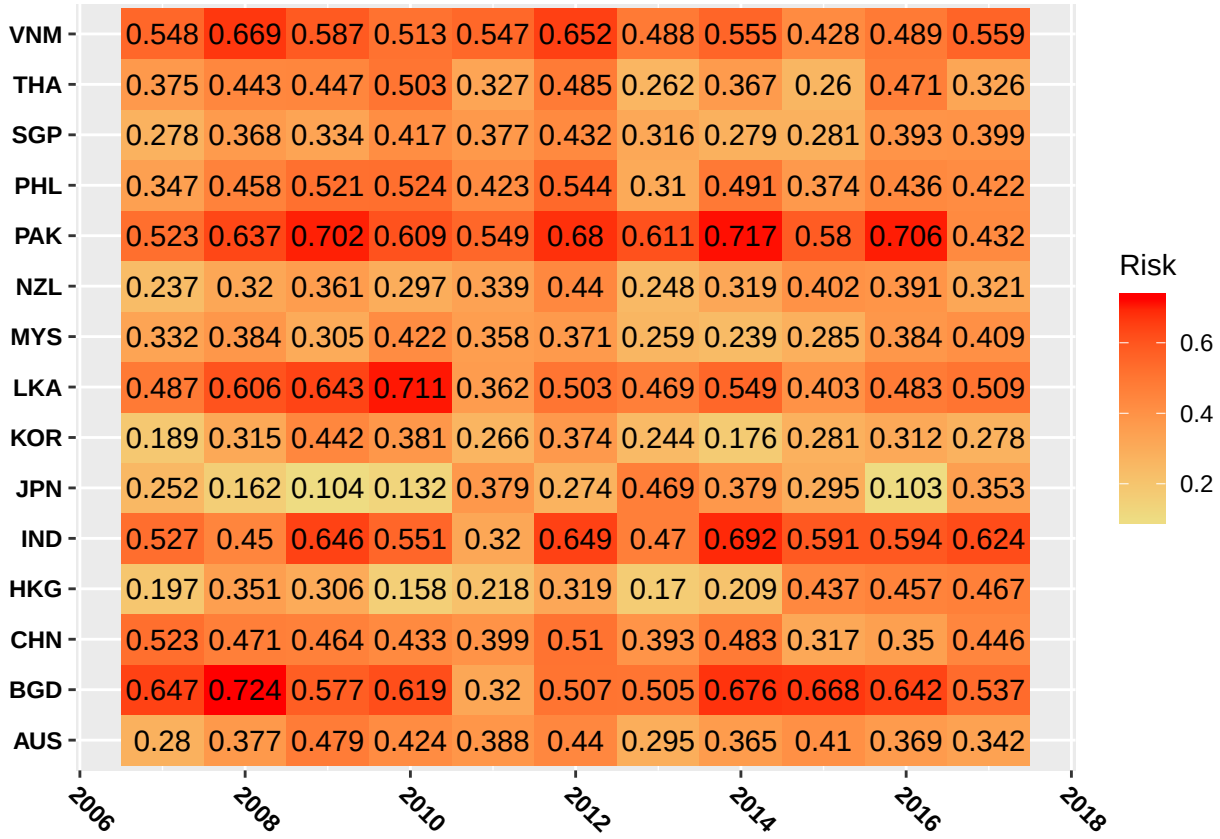
4 Plots

4.1 Heatmap

```
# Heat Map
ggplot(data = index, aes(x = year, y = iso3c)) +
  geom_tile(aes(fill = Risk)) +
  geom_text(aes(label = round(Risk, 3))) +
  scale_fill_gradient(low = 'lightgoldenrod2',high = 'red') +
  theme(axis.text.x = element_text(angle = -45, vjust = 1, hjust = 0)) +
  theme(axis.title.x=element_blank()) +
  theme(axis.title.y=element_blank()) +
  theme(axis.text.x = element_text(color = 'black', face="bold")) +
```

```
  theme(axis.text.y = element_text(color = 'black', face="bold")) +
  ggsave("RiskIndex_Heatmap.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

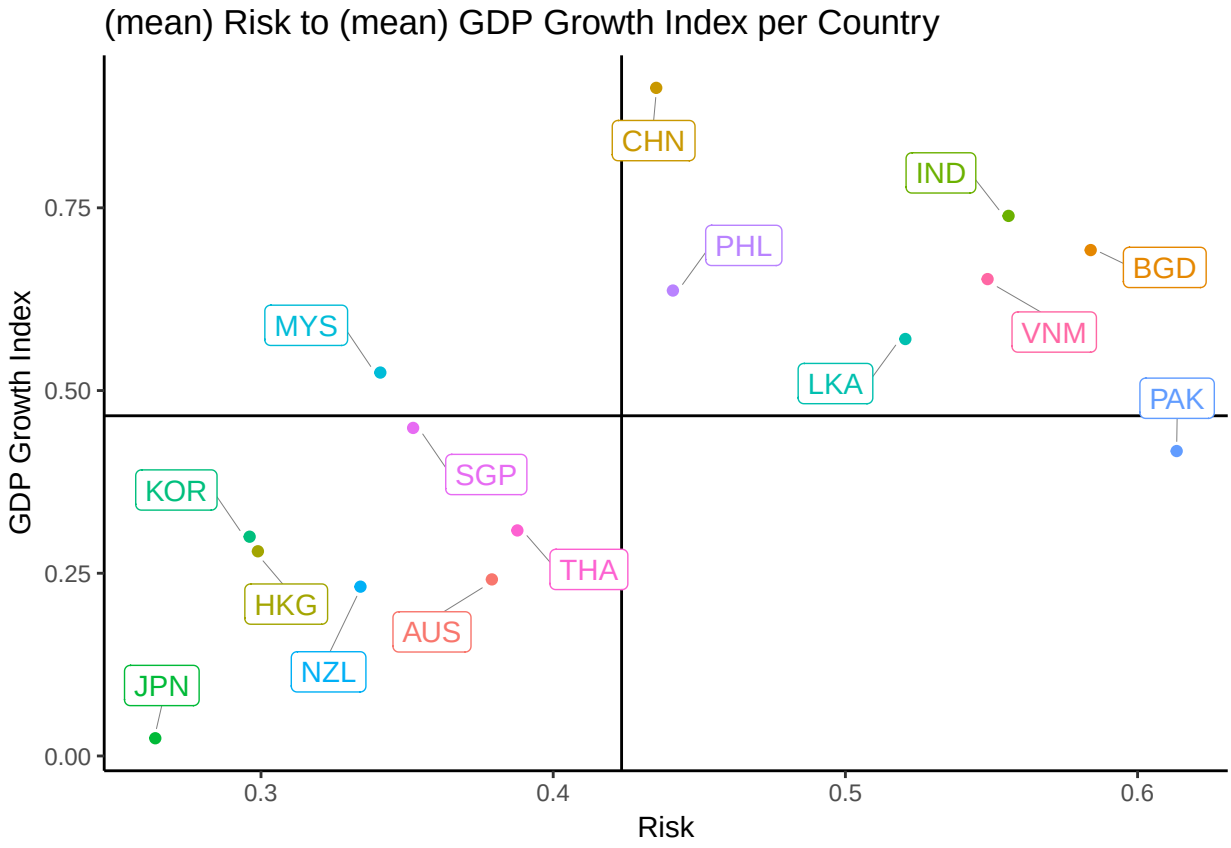


4.2 Scatter Plot

```
# Create Plot (mean) Risk to (mean) GDP Growth Index per Country
index %>%
  group_by(iso3c) %>%
  summarise(meanr = mean(Risk), meangdp = mean(`g_index`)) %>%
  ggplot(aes(x = meanr, y = meangdp, label = iso3c, colour = iso3c)) +
  geom_point() +
  geom_vline(aes(xintercept = mean_Risk)) +
  geom_hline(aes(yintercept = mean_GDPindex)) +
  labs(title = "(mean) Risk to (mean) GDP Growth Index per Country",
       y = "GDP Growth Index",
       x = "Risk") +
  geom_label_repel(aes(label = iso3c),
                   box.padding   = 0.35,
                   point.padding = 0.5,
                   segment.color = 'grey50',
                   segment.size  = 0.1) +
```

```
  theme_classic() +
  theme(legend.position="none") +
  ggsave("gindexrisk_scatter.pdf")
```

```
## Saving 6.5 x 4.5 in image
```



5 References

- World Bank
  - Domestic credit to private sector (% of GDP)
  - S&P Global Equity Indices (annual % change)
  - Foreign direct investment, net (BoP, current US$)
  - Portfolio investment, net (BoP, current US$)
  - Regulatory Quality: Estimate
  - Inflation, consumer prices (annual %)
  - Bank non-performing loans to gross loans (%) I
  - Bank non-performing loans to gross loans (%) II
  - GDP growth (annual %)
  - GDP (current US$)
  - Official exchange rate (LCU per US$, period average)
- Federal Reserve Bank of St. Louis
  - Bank Non-Performing Loans to Gross Loans for Bangladesh
  - Bank Non-Performing Loans to Gross Loans for Japan

  - Bank Non-Performing Loans to Gross Loans for Singapore
  - Bank Non-Performing Loans to Gross Loans for Pakistan
  - Bank Non-Performing Loans to Gross Loans for Indonesia
  - Bank Non-Performing Loans to Gross Loans for South Korea
  - Bank Non-Performing Loans to Gross Loans for Australia
  - Bank Non-Performing Loans to Gross Loans for China
  - Bank Non-Performing Loans to Gross Loans for Hong Kong, SAR
  - Bank Non-Performing Loans to Gross Loans for Philippines
  - Bank Non-Performing Loans to Gross Loans for Malaysia
  - Bank Non-Performing Loans to Gross Loans for Thailand
- Principal Component Analysis I
- Principal Component Analysis II
- Principal Component Analysis III
- Principal Component Analysis IV