##Loading the Packages which are required

#Relevant Functions used in the Code

```
"Correlation Heatmap Function"
```

```
## [1] "Correlation Heatmap Function"
```

```
get_heatmap = function(dat,columns,col_name){
  corr = round(x = cor(dat[columns],use = "complete.obs"), digits = 2)
  melted_corr = reshape2::melt(corr)
  heatplot = ggplot(data = melted_corr, aes(x=Var1, y=Var2, fill = value)) +
    geom_tile(color = "white")+
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
    midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle=45, hjust=1)) +
    labs(title = paste0("Correlation Heatmap for ",col_name))+
    coord_fixed()
  ggplotly(heatplot)
}

"Multiple Line chart Function to draw line chart across time with the Macro economic variables"
```

```
## [1] "Multiple Line chart Function to draw line chart across time with the Macro economic variables"
```

```
#Plot the multiple line charts for the securities along with the macro-economic indicators
plot_multiple_line_chart = function(dat1, title_type,y1_type,macro_flag = 0) {
  if(macro_flag == 0){
    cols_taken = ncol(dat1)
  }else{
    cols_taken = ncol(dat1)-1
  }
  plot_y <- dat1 %>% plot_ly()
  for(i in c(2:ncol(dat1))) {
    if (i <= cols_taken) {
      x = plot_y %>% add_trace(x = ~fyear, y=dat1[[i]],  mode="lines" ,type = 'scatter', name=colnames(dat1)[i],
yaxis='y1')
    } else if(macro_flag != 0){
      x = plot_y %>% add_trace(x = ~fyear, y=dat1[[i]], mode="lines", type = 'scatter', name=colnames(dat1)[i], y
axis='y2')
    }
    plot_y = x
  }
  if(macro_flag != 0)
  {
      plot_y %>%
      layout(title = paste0(title_type,"(LHS) vs Macro economic variable (RHS)"),
         barmode   = 'relative',
         xaxis     = list(title=''),
         margin    = list(l = 75, r = 75, b = 50, t = 50, pad = 4),
         xaxis     = list(title = ""),
         yaxis     = list(side = 'left',
                          title = y1_type,
                          showgrid = FALSE,
                          zeroline = TRUE,
                          color = 'steelblue'),
         yaxis2    = list(side = 'right',
                          overlaying = "y",
                          title = colnames(dat1)[ncol(dat1)],
                          showgrid = TRUE,
                          zeroline = FALSE,
                          # ticksuffix = "%",
                          color = "#ffa500"),
         legend = list(traceorder = 'reversed',orientation = "h"))
  }
  else{
    plot_y %>%
    layout(title = paste0(title_type," Data"),
       barmode   = 'relative',
       xaxis     = list(title=''),
       margin    = list(l = 75, r = 75, b = 50, t = 50, pad = 4),
       xaxis     = list(title = ""),
       yaxis     = list(side = 'left',
                          title = y1_type,
                          showgrid = FALSE,
                          zeroline = TRUE,
                          color = 'steelblue'),
       legend = list(traceorder = 'reversed',orientation = "h"))
  }
}

"Box-Plot Function"
```

```
## [1] "Box-Plot Function"
```

```
get_boxplot = function(in_data,variable_name,full_plot){
  box_plot_data = in_data%>%
    select(fyear,variable_name)
  names(box_plot_data)[2] = "value"

  if(full_plot == 1)
  {
    p = ggplot(box_plot_data, aes(x=as.factor(fyear), y=value)) +
      geom_boxplot(fill='red', color="dark green")+
      theme_classic()+
      coord_cartesian(y = c(-5000,5000))+
      labs(x = "Year", y = variable_name)
  }else{
      p = ggplot(box_plot_data, aes(x=as.factor(fyear), y=value)) +
        geom_boxplot(fill='red', color="dark green")+
        theme_classic()+
        coord_cartesian(y = c(-15,15))+
        labs(x = "Year", y = variable_name)
  }
  return (p)
}

"Box-Plots for all variable(Loop)"
```

```
## [1] "Box-Plots for all variable(Loop)"
```

```
plot_all_boxplots = function(in_data,full = 0){
  col_names = colnames(in_data)[2:ncol(in_data)]
  for (i in 1:length(col_names)){
    print(get_boxplot(in_data,col_names[i],full))
  }
}

"Descriptive Stats Function : count, mean, p25, p50, p75, std, max, min"
```

```
## [1] "Descriptive Stats Function : count, mean, p25, p50, p75, std, max, min"
```

```
stats = function(dat){
count_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(),funs(sum(!is.na(.))),.names = "count_{.col}"))
mean_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), mean,na.rm = TRUE,.names = "mean_{.col}"))
p25_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), quantile,probs = c(0.25),na.rm = TRUE,.names = "p25_{.col}"))
p50_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), quantile,probs = c(0.5),na.rm = TRUE,.names = "p50_{.col}"))
p75_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), quantile,probs = c(0.75),na.rm = TRUE,.names = "p75_{.col}"))
std_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), sd,na.rm = TRUE,.names = "std_{.col}"))
max_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), max,na.rm = TRUE,.names = "max_{.col}"))
min_group = dat %>%
  group_by(fyear)%>%
  summarise(across(everything(), min,na.rm = TRUE,.names = "min_{.col}"))
 return (list(count_group,mean_group, p25_group,p50_group, p75_group, std_group,max_group,min_group))
}
```

##Reading the data

```
"Selecting the variables to read from the SAS dataset (Combined the variables from excel sheet)"
```

```
## [1] "Selecting the variables to read from the SAS dataset (Combined the variables from excel sheet)"
```

```
#Reading the variables extracted from the tables
variables = read.csv("Variable_Names.csv")
variables$lower = tolower(variables$Variable_Name)
variable_list = unique(variables$lower)
```

##Dateset Manipulation

```
"Importing the dataset with only selected variables"
```

```
## [1] "Importing the dataset with only selected variables"
```

```
# path = "Q:/Data-ReadOnly/COMP/"
# subset_data = read_sas(paste0(path,"funda.sas7bdat"),n_max = 1000)
col_list = c("gvkey","fyear","indfmt","tic","conm","scf","compst","sich",variable_list)
subset_data = read_sas("funda.sas7bdat",col_select = col_list)

"Filtering the dataset based on the requirements"
```

```
## [1] "Filtering the dataset based on the requirements"
```

```
#Taking only the relevant data and dropping rows corresponding to the column Total Assets which have NA values. N
o point of taking the other values in the row (variables) if we don't have Total assets(at) data
subset_data = subset_data%>%
  filter(compst != "AB",!(sich %in% 4900:4999),!(sich %in% 6000:6999),!(scf %in% (4:6)))%>%
  select(-c(compst,sich))%>%
  drop_na(at,scf)
```

##GDP Defaltor loaded

```
"For every year multiplier is calculated based on the base year 1992(67.89 value of the GDP Defaltor)"
```

```
## [1] "For every year multiplier is calculated based on the base year 1992(67.89 value of the GDP Defaltor)"
```

```
gdp_deflator <-
   "GDPDEF" %>%
  tq_get(get = "economic.data", from = "1971-01-01") %>%
  select(fyear = date, index  =  price) %>%
  mutate(fyear = as.numeric(year(fyear)))%>%
  group_by(fyear)%>%
  slice(n())%>%
  mutate(multiplier = 67.889/index)
```

##Subset Data after 1992 dollar terms

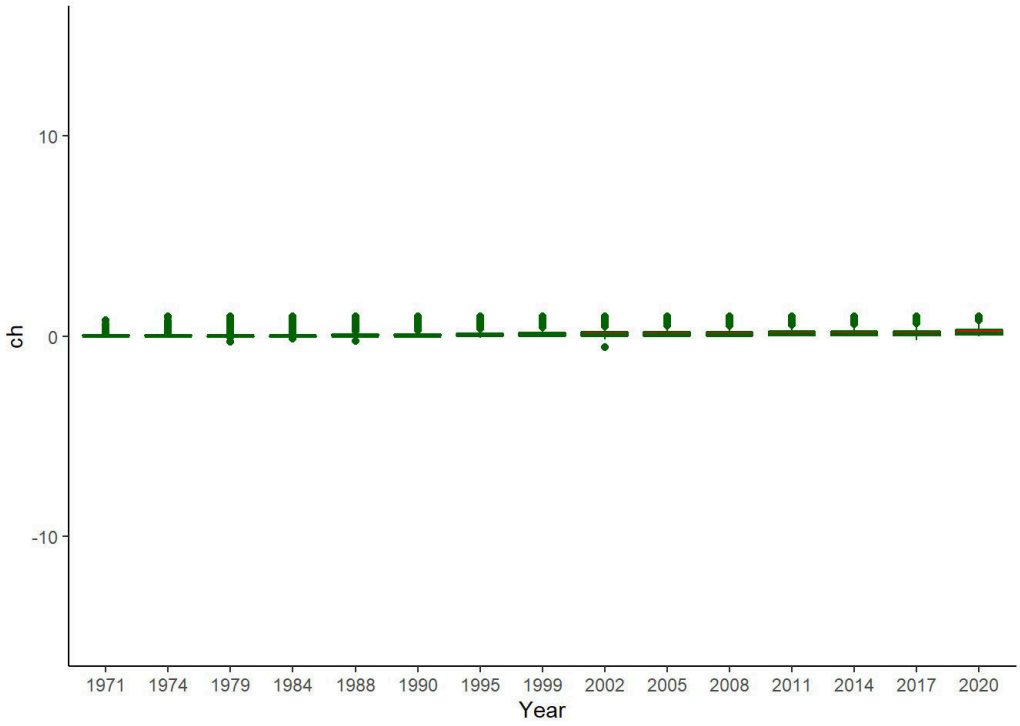
```
#Final data which can be used for further analysis
year_list = c(1971,1974,1979,1984,1988,1990,1995,1999,2002,2005,2008,2011,2014,2017,2020)
selected_data = subset_data%>%
  filter(fyear %in% year_list)%>%
  merge(gdp_deflator[c("fyear","multiplier")],by = "fyear")

selected_data[,7:(ncol(selected_data)-1)] = selected_data[,7:(ncol(selected_data)-1)]*unlist(selected_data["multi
plier"])
selected_data = selected_data%>%
  select(-multiplier)
```

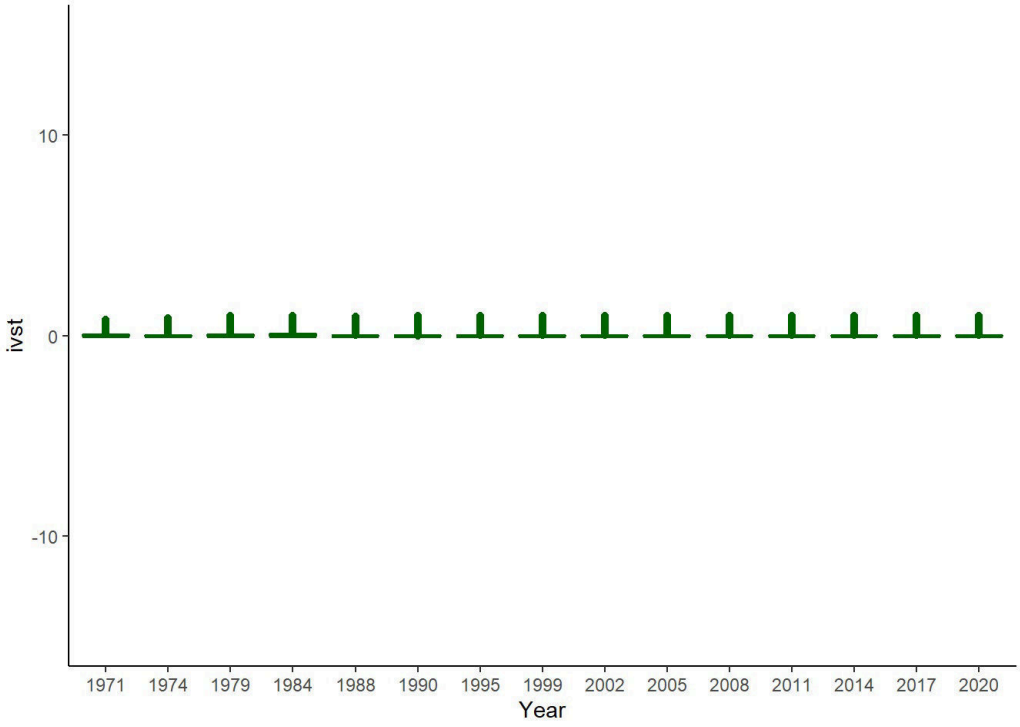
##Desciptive Stats and Box Plots of tables

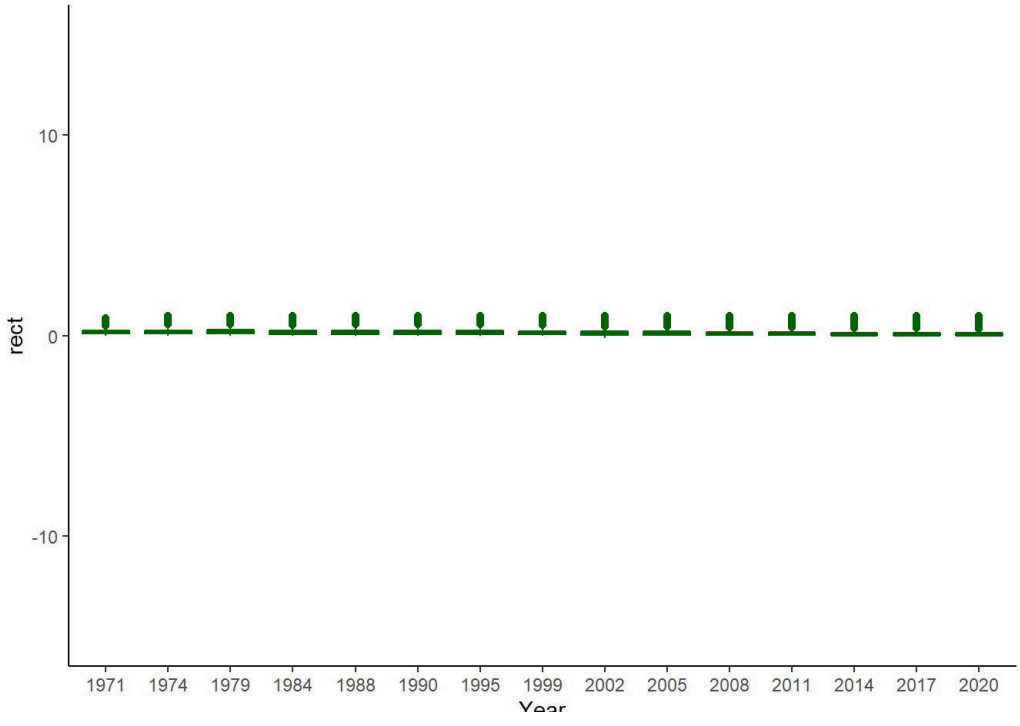
```
"Modifying the tables and recording the original NA values, calculating the descriptive stats and box plots for a
ll variables in the table"
```

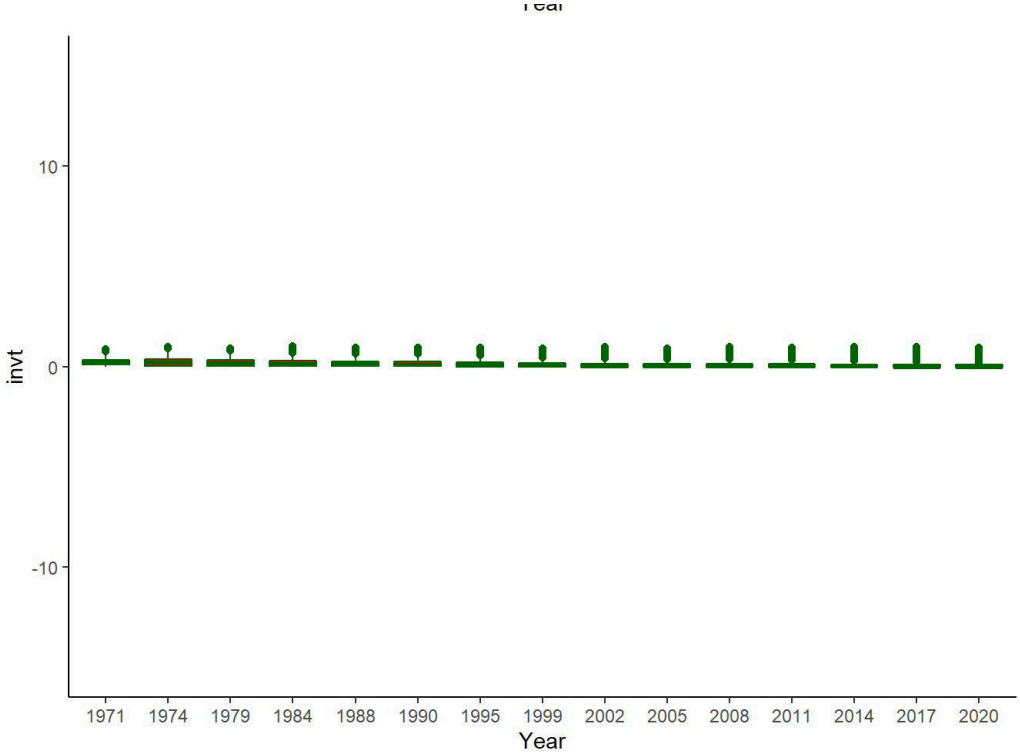
```
## [1] "Modifying the tables and recording the original NA values, calculating the descriptive stats and box plot
s for all variables in the table"
```

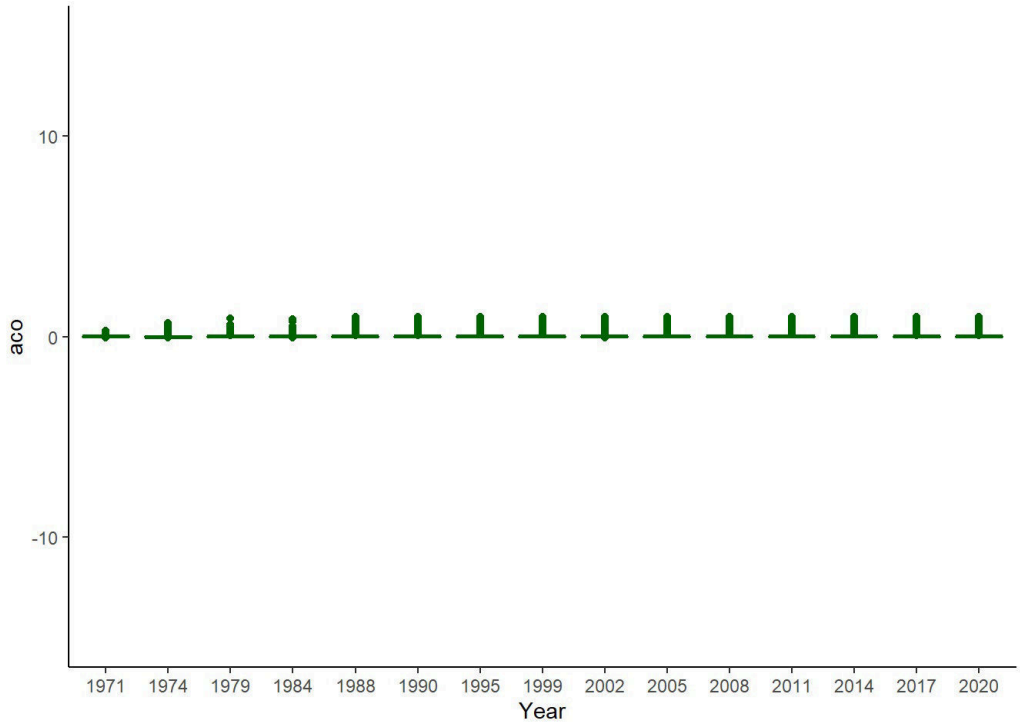
```
#Taking only those rows for a year of a company where total assets are not zero
selected_data = selected_data[selected_data["at"]!=0,]
selected_data = arrange(selected_data,gvkey,fyear)

na_values = is.na(selected_data)
#Replacing remaining NA values with zero so, as to compute the compound variables
selected_data[is.na(selected_data)] = 0

Modified_Data = selected_data %>%
  mutate(non_operating_income_special_items = nopi+spi,
         other_funds_from_operation = xidoc+txdc+esubc+sppiv+fopo,
         short_term_investment = ivstch+ivaco)

#Copying the NA's back to the original data so that we don't include the balance sheet variables in the calculati
on of mean etc.
Modified_Data_NA = copy(Modified_Data)
for (i in (1:ncol(na_values))){
  if(any(na_values[,i])){
    Modified_Data_NA[,i] = replace(as.vector(unlist(Modified_Data_NA[,i])), na_values[,i] == TRUE, NA)
  }
}

balance_sheet_variables = c("ch","ivst","rect","invt","aco","act","ppent","ivaeq","ivao","intan","ao","at",
                            "dlc","ap","txp","lco","lct","dltt","lo","txditc","mib","lt","pstk","ceq","teq",
                            "at")
income_statement_variables = c("sale","cogs","xsga","oibdp","dp","oiadp","xint","non_operating_income_special_ite
ms","pi","txt","mii","ib","dvp","cstke","xido","ni")

cash_flow_statement_variables = c("ibc","dpc","other_funds_from_operation","fopt","recch","invch","apalch","txach
","aoloch","oancf","ivch","siv","capx","sppe","aqc","short_term_investment","ivncf","sstk","prstkc","dv","dltis",
"dltr","dlcch","fiao","fincf","exre","chech","fsrco","fuseo","wcapc")

Balance_Sheet_Data = Modified_Data_NA%>%
  select(c(fyear,balance_sheet_variables))%>%
  group_by(fyear)%>%
  mutate(across(everything(),.fns = ~./at))

Income_Statement_Data = Modified_Data_NA%>%
  select(c(fyear,income_statement_variables,at))%>%
  group_by(fyear)%>%
  mutate(across(everything(),.fns = ~./at))%>%
  select(-at)

Cash_Flow_Data = Modified_Data_NA%>%
  select(c(fyear,cash_flow_statement_variables,at))%>%
  group_by(fyear)%>%
  mutate(across(everything(),.fns = ~./at))%>%
  select(-at)

#Descriptive Stats for the tables
c(count_balance_sheet_data,mean_balance_sheet_data,p25_balance_sheet_data,p50_balance_sheet_data, p75_balance_she
et_data, std_balance_sheet_data,max_balance_sheet_data,min_balance_sheet_data)%<-%stats(Balance_Sheet_Data)

c(count_income_statement_data,mean_income_statement_data,p25_income_statement_data,p50_income_statement_data, p75
_income_statement_data, std_income_statement_data,max_income_statement_data,min_income_statement_data)%<-%stats(I
ncome_Statement_Data)


c(count_cash_flow_data,mean_cash_flow_data,p25_cash_flow_data,p50_cash_flow_data, p75_cash_flow_data, std_cash_fl
ow_data,max_cash_flow_data,min_cash_flow_data)%<-%stats(Cash_Flow_Data)
#Box plots of all variables in the three tables(Balance Sheet, Income Statement, Cash Flow Statement)
# pdf("All_plots.pdf")
plot_all_boxplots(Balance_Sheet_Data)
```

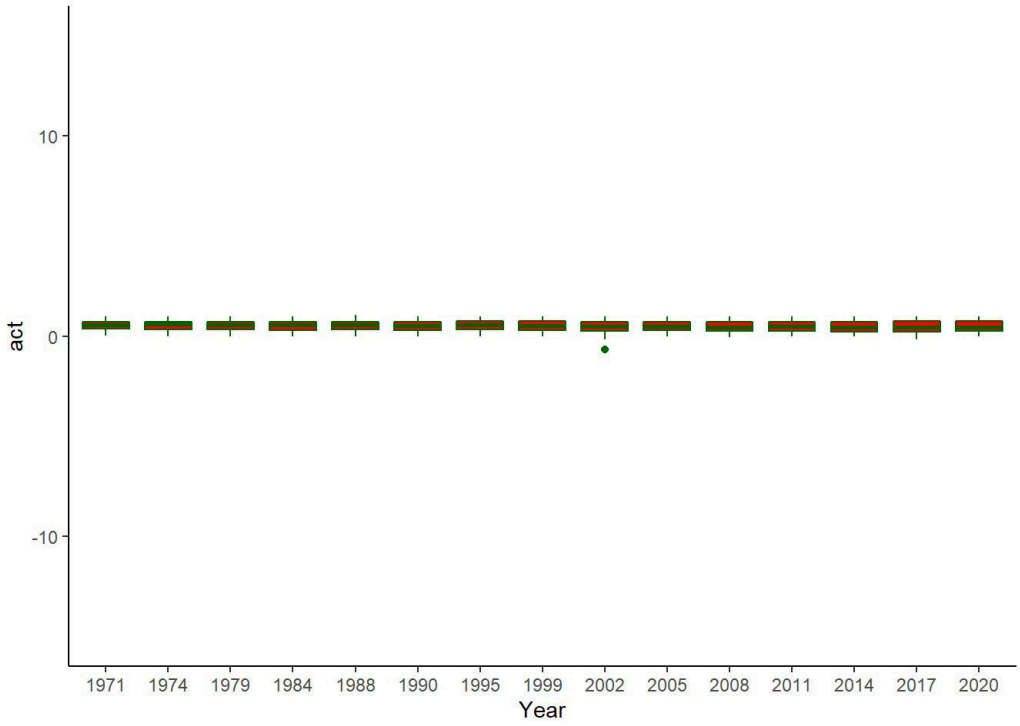

ch
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

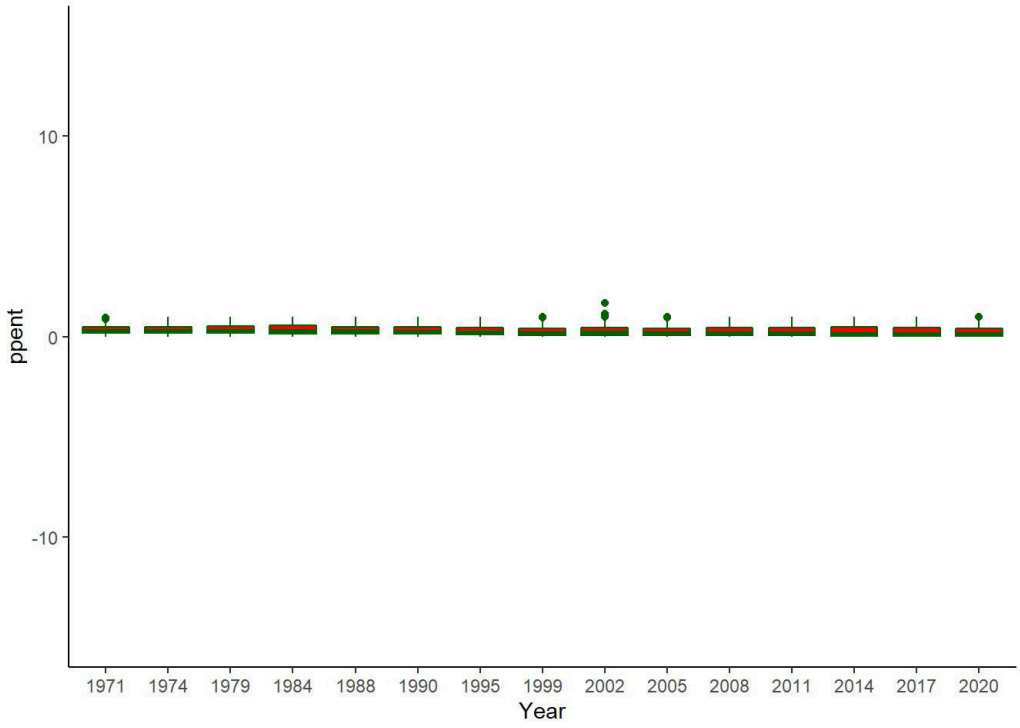

ivst
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

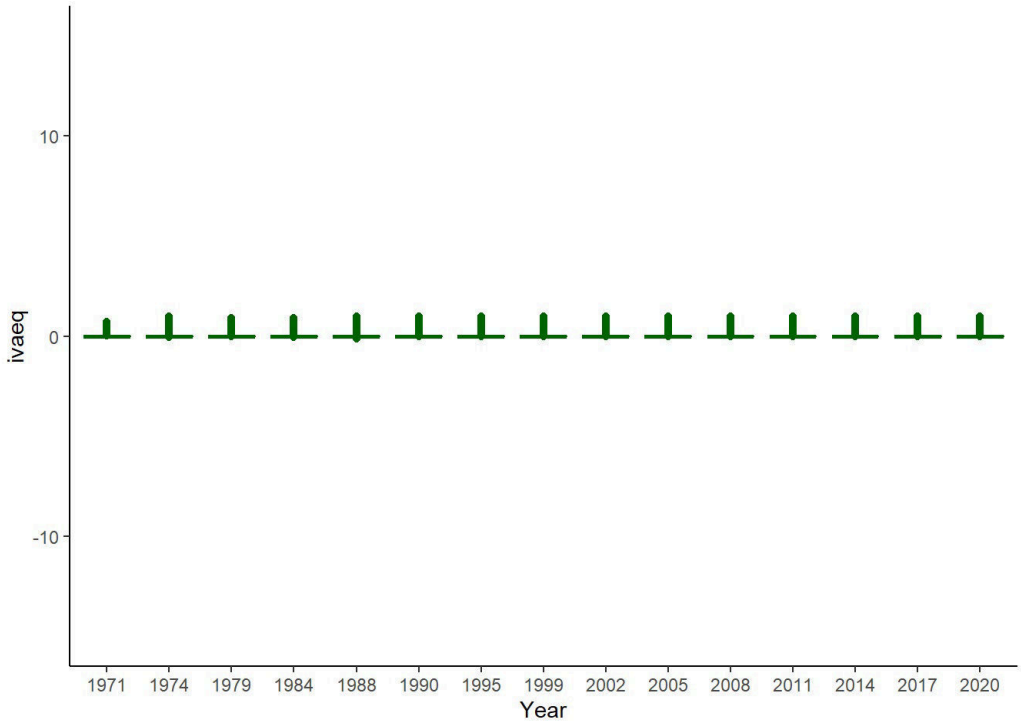

rect
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

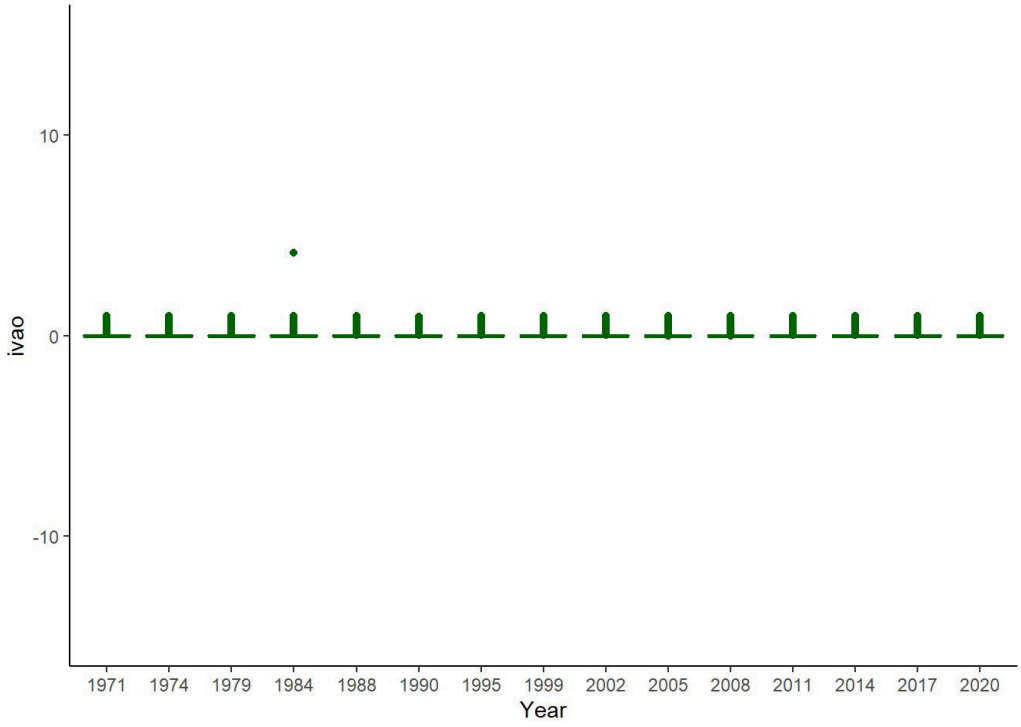

invt
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

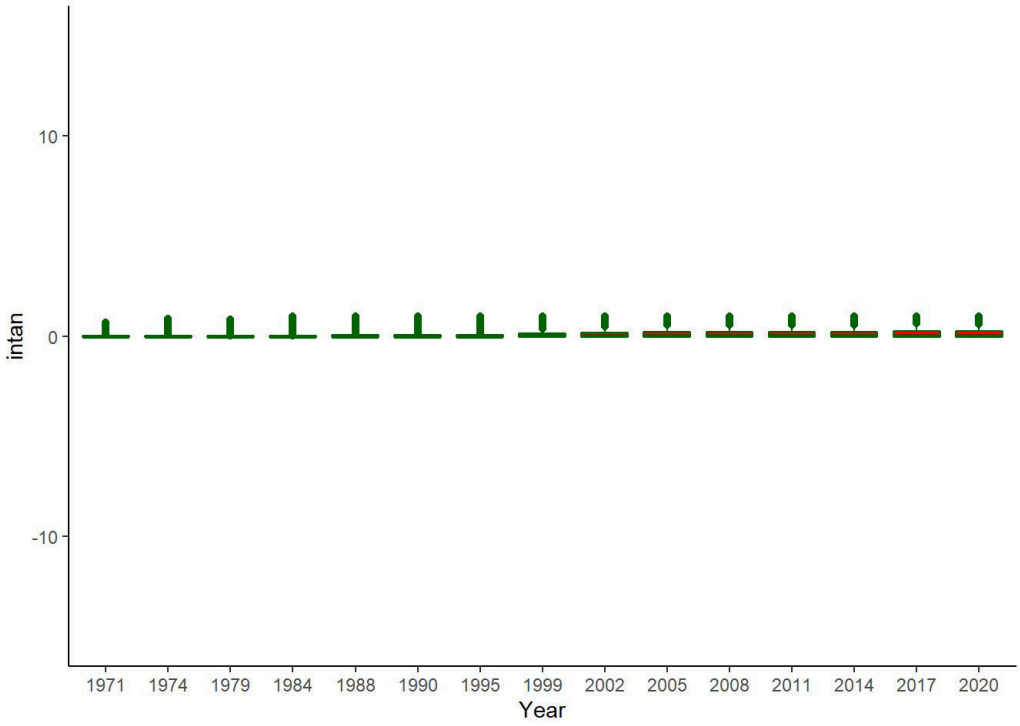

aco
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

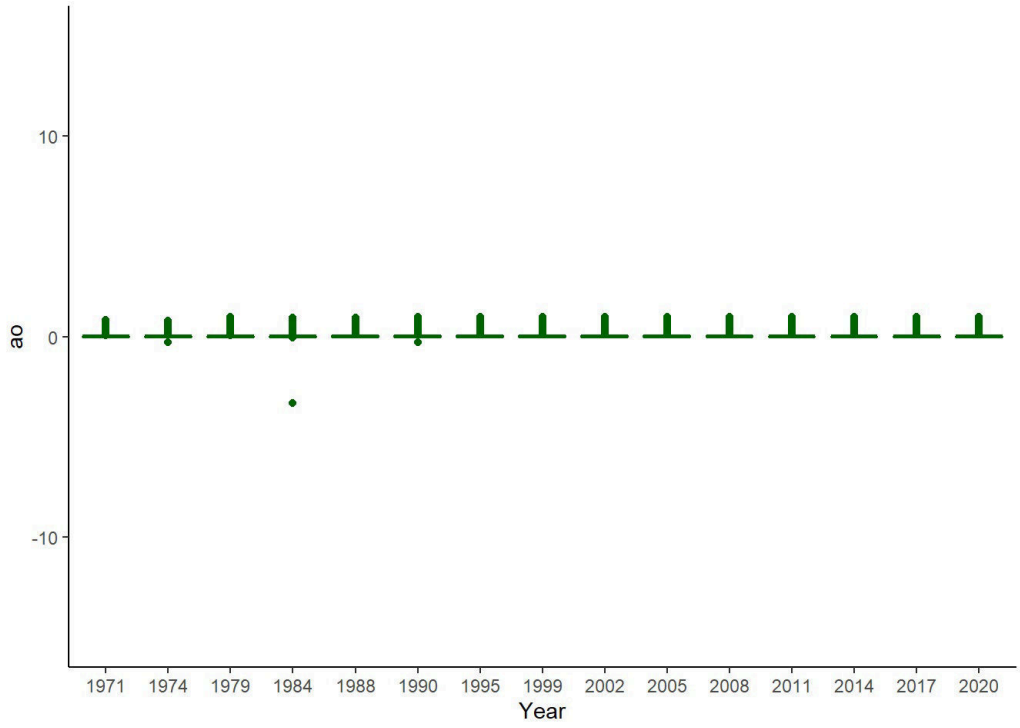

act
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

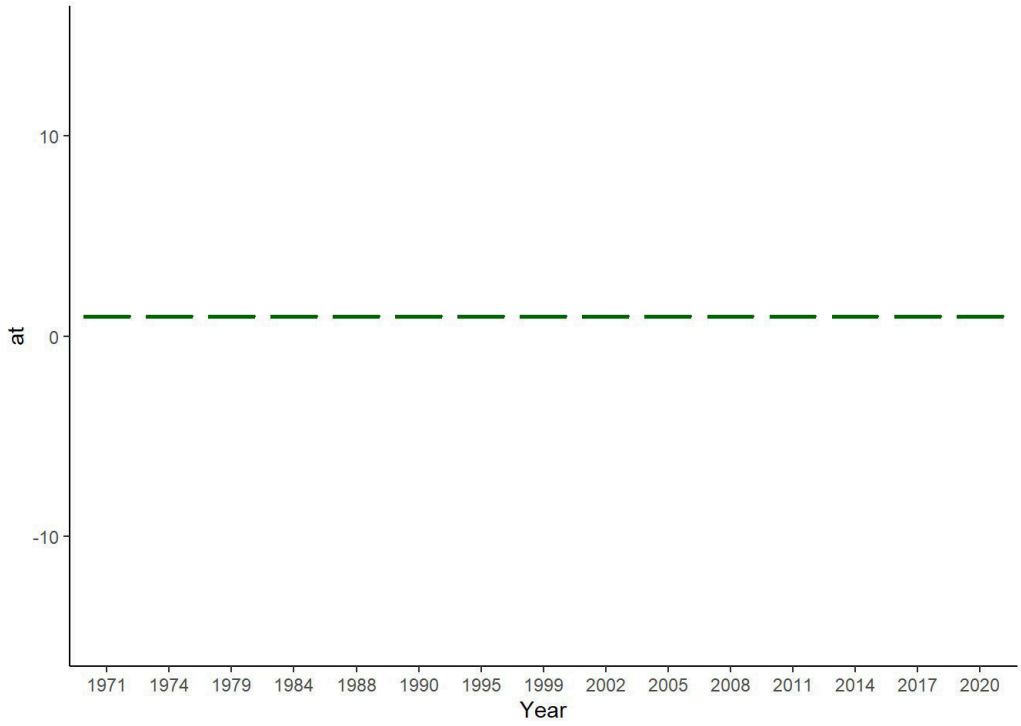

ppent
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

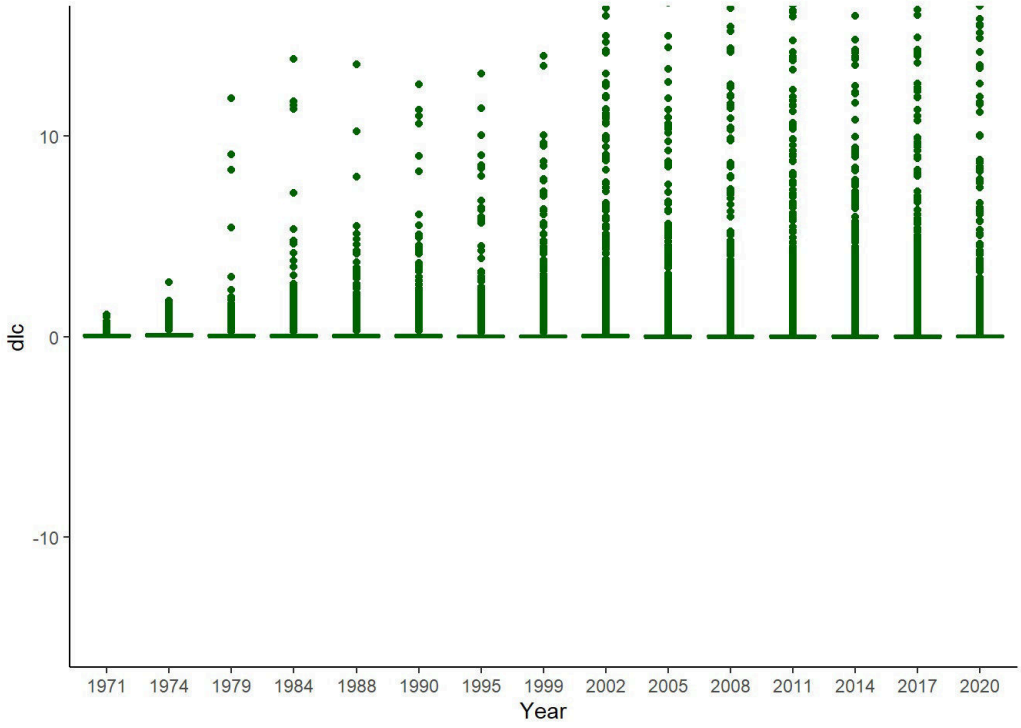

ivaeq
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

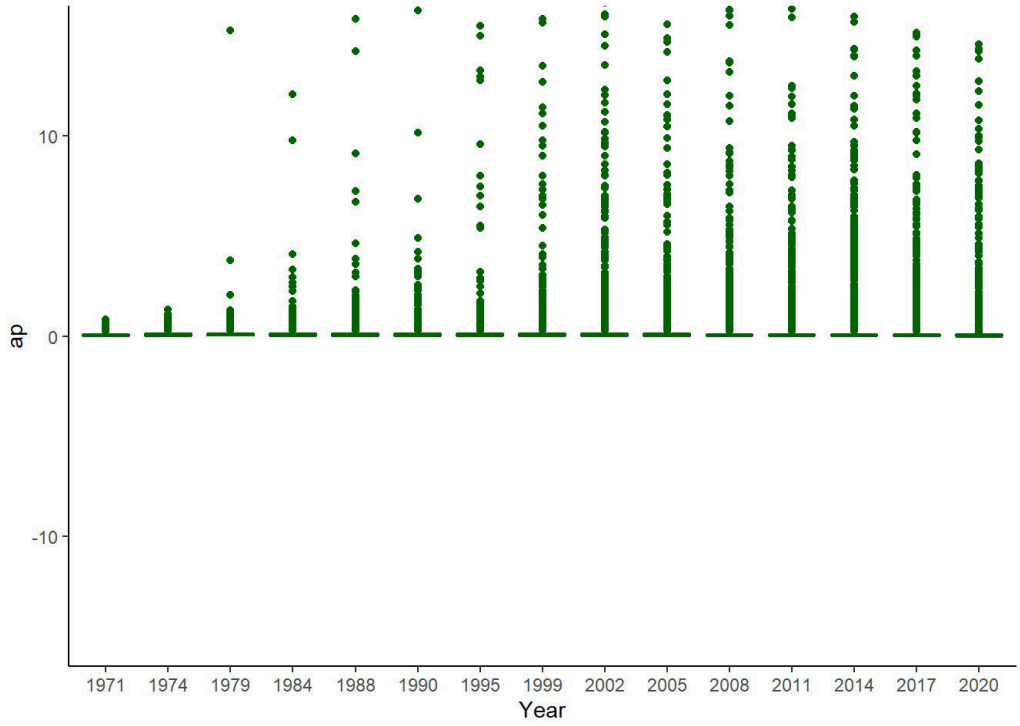

ivao
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

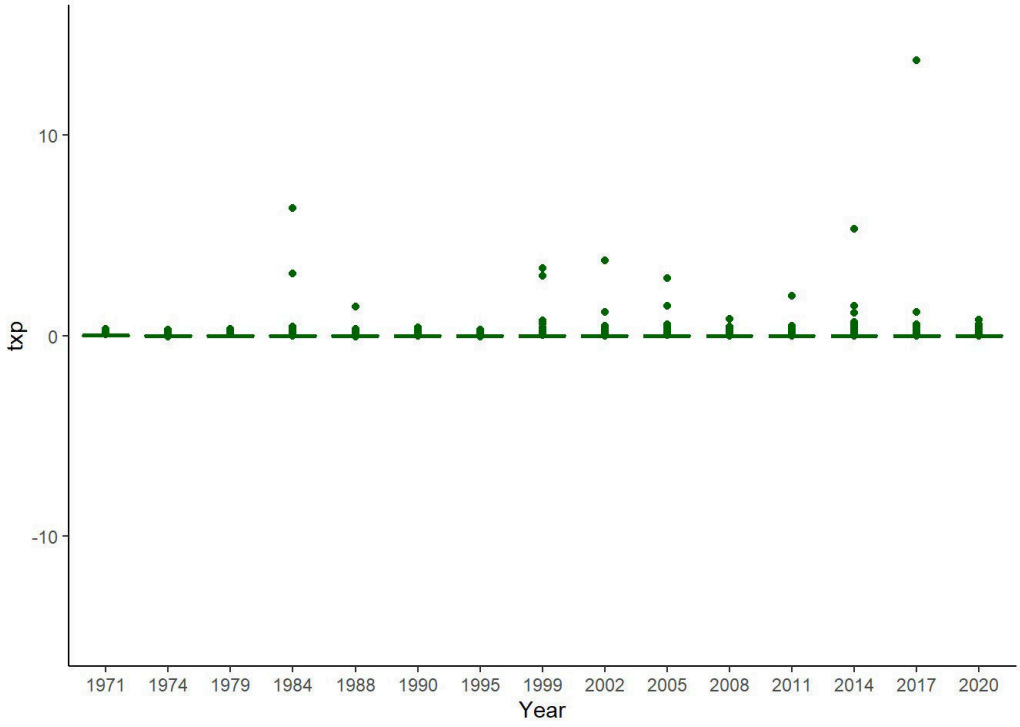

intan
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

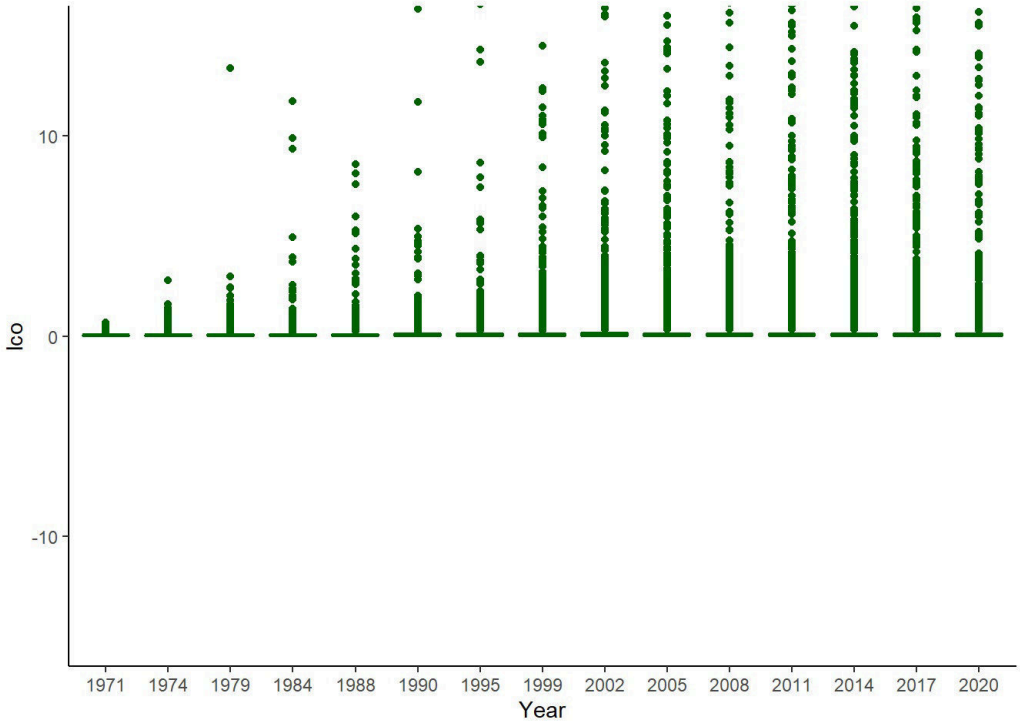

ao
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

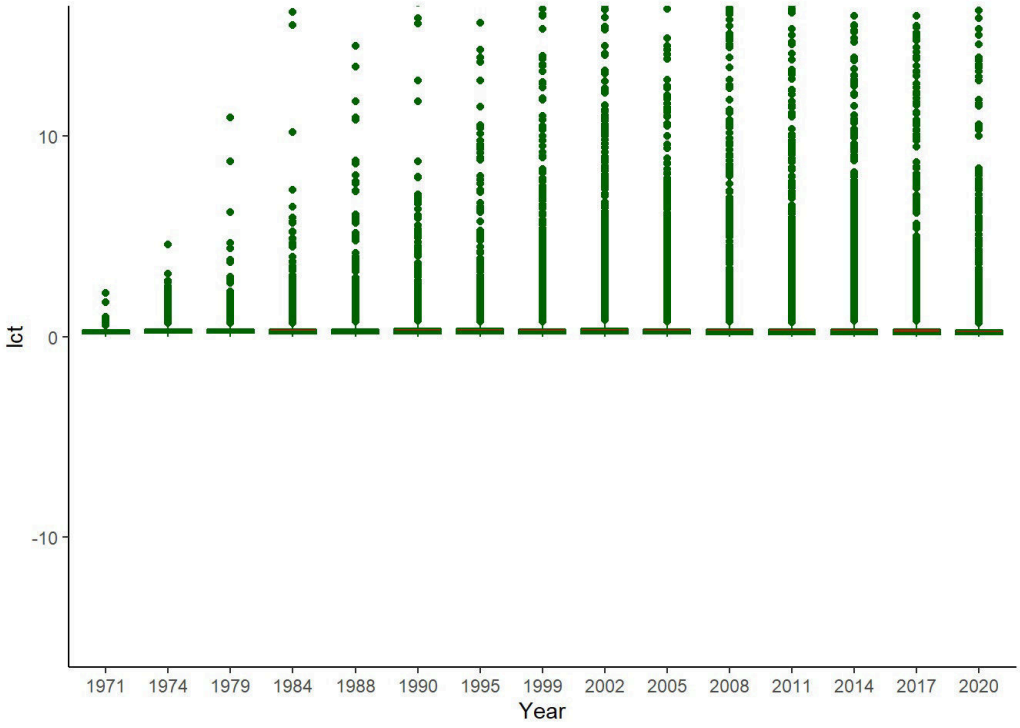

10
0
-10
at
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

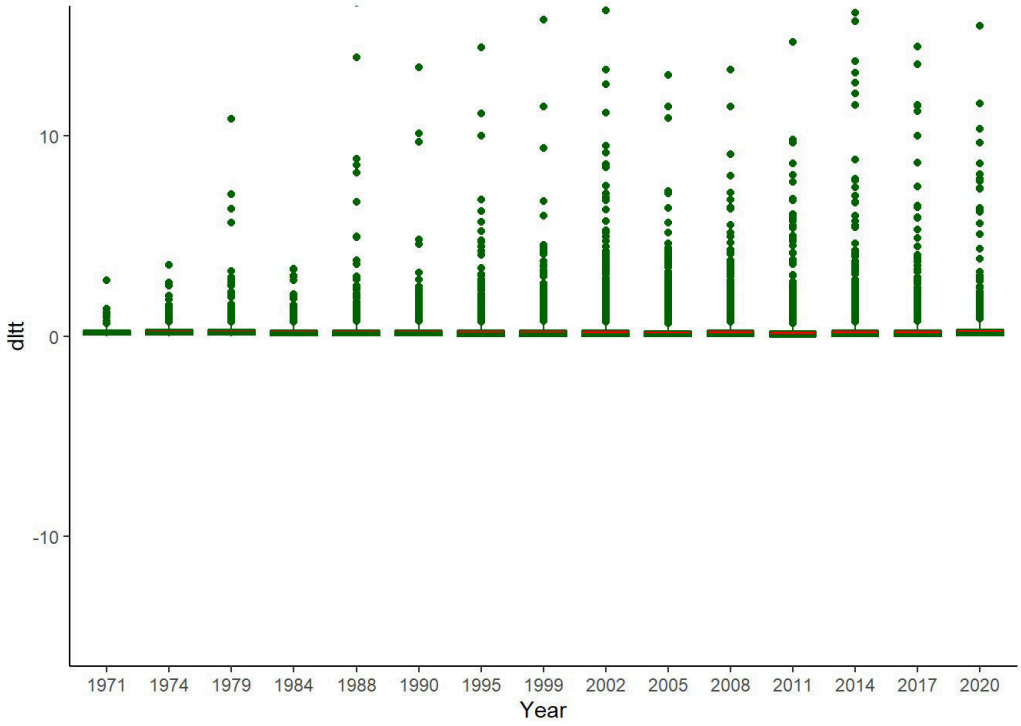

10
0
-10
dlc
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

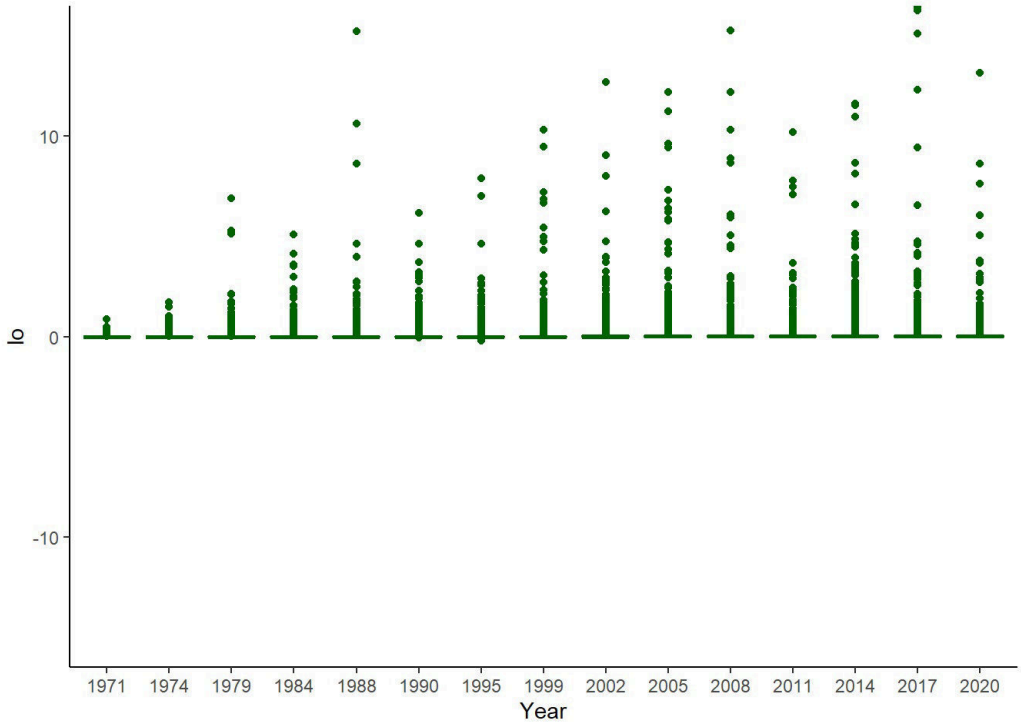

ap
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

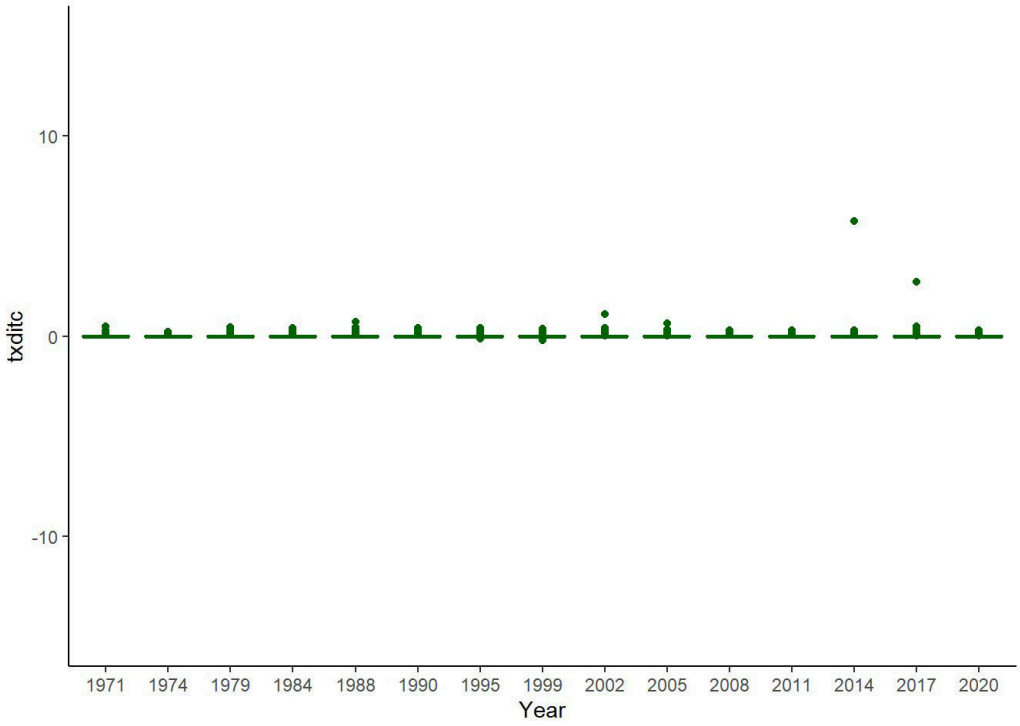

txp
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

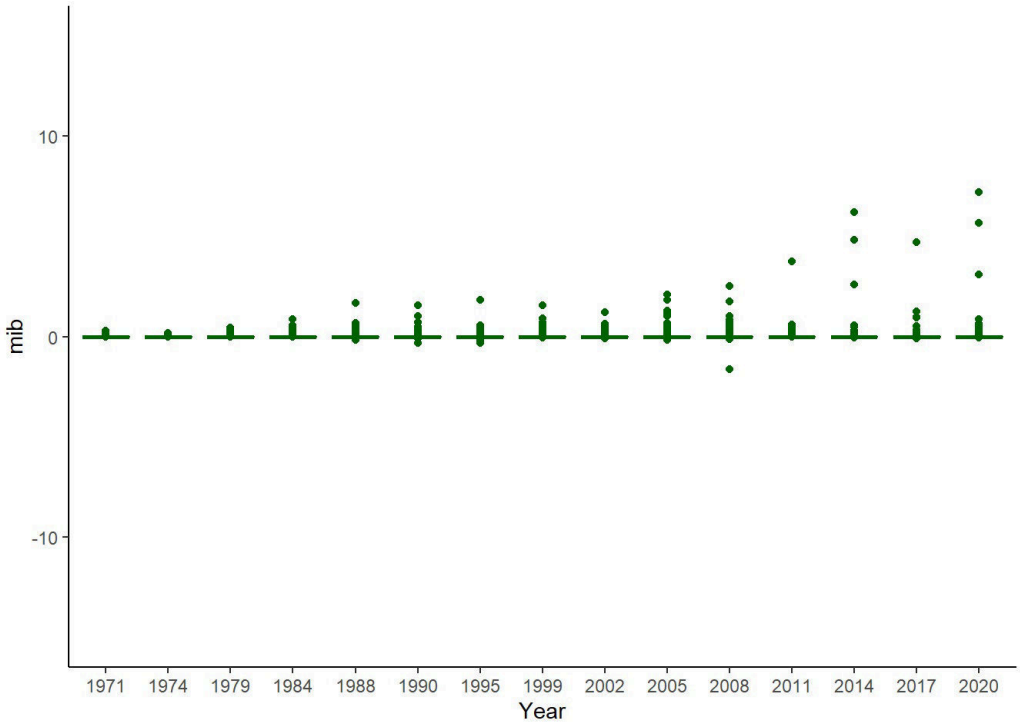

lco
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

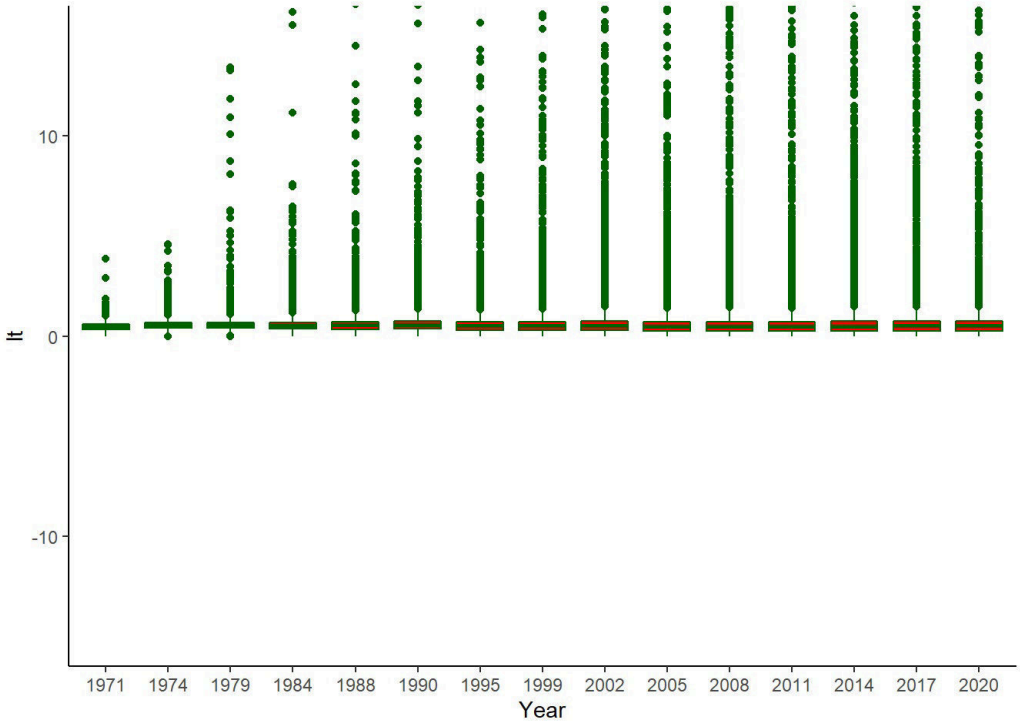

lct
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

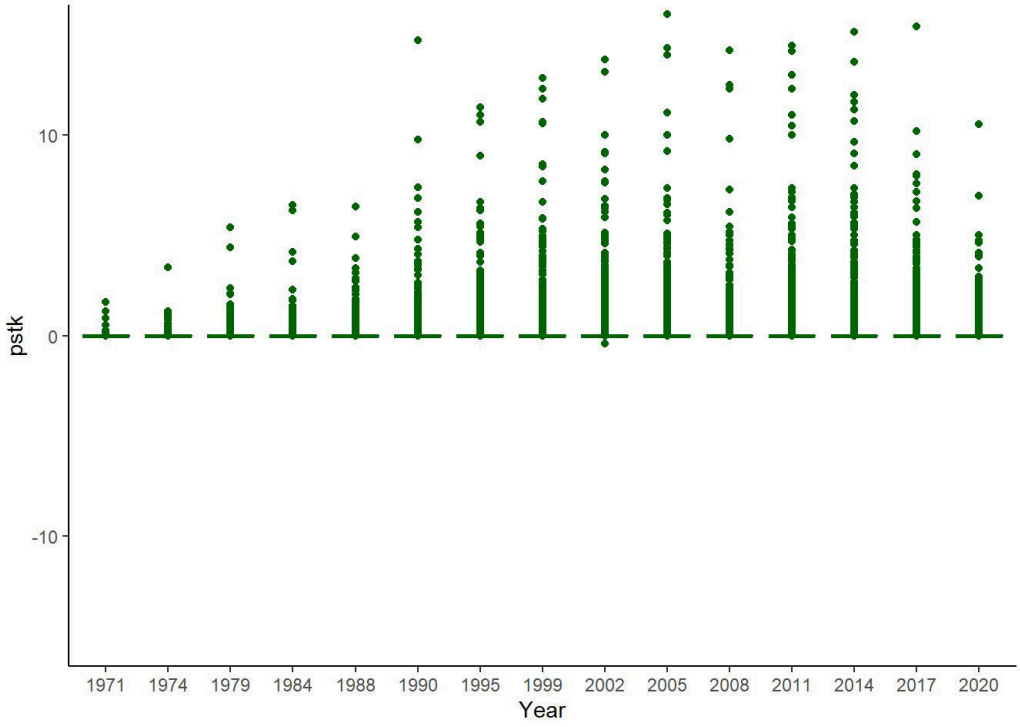

dltt
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

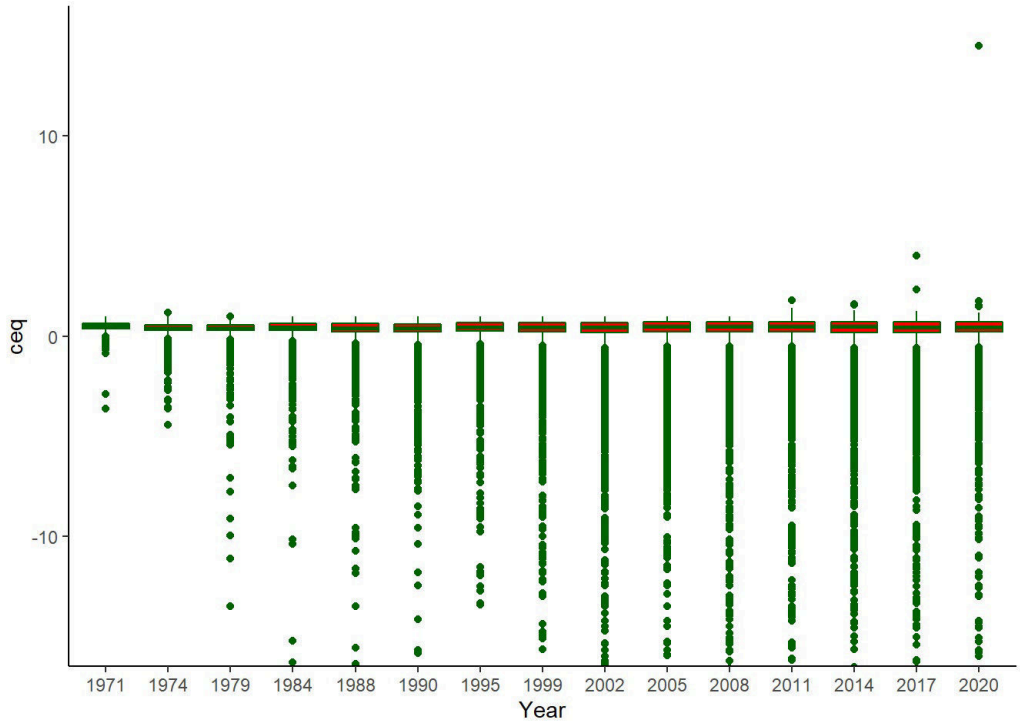

lo
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

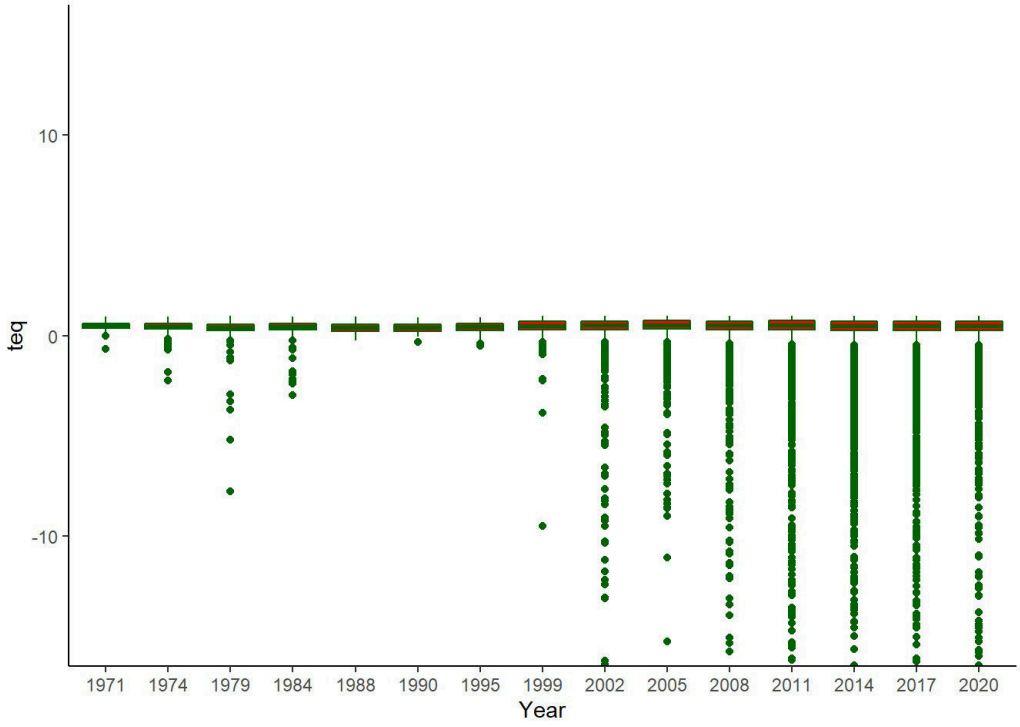

txditc
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


mib
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

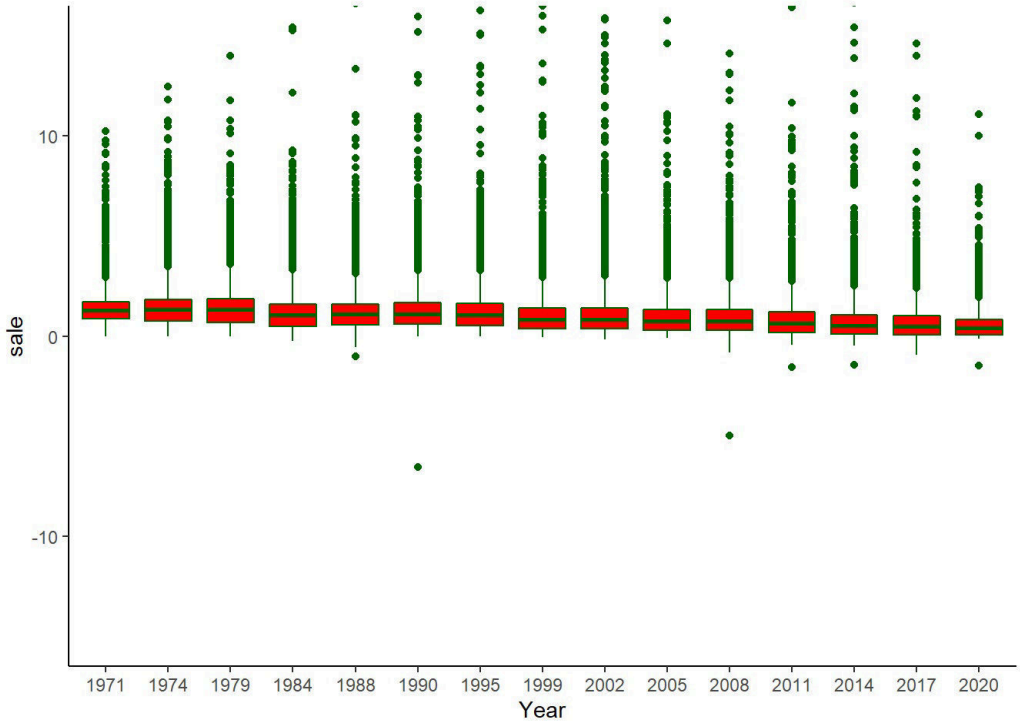

lt
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

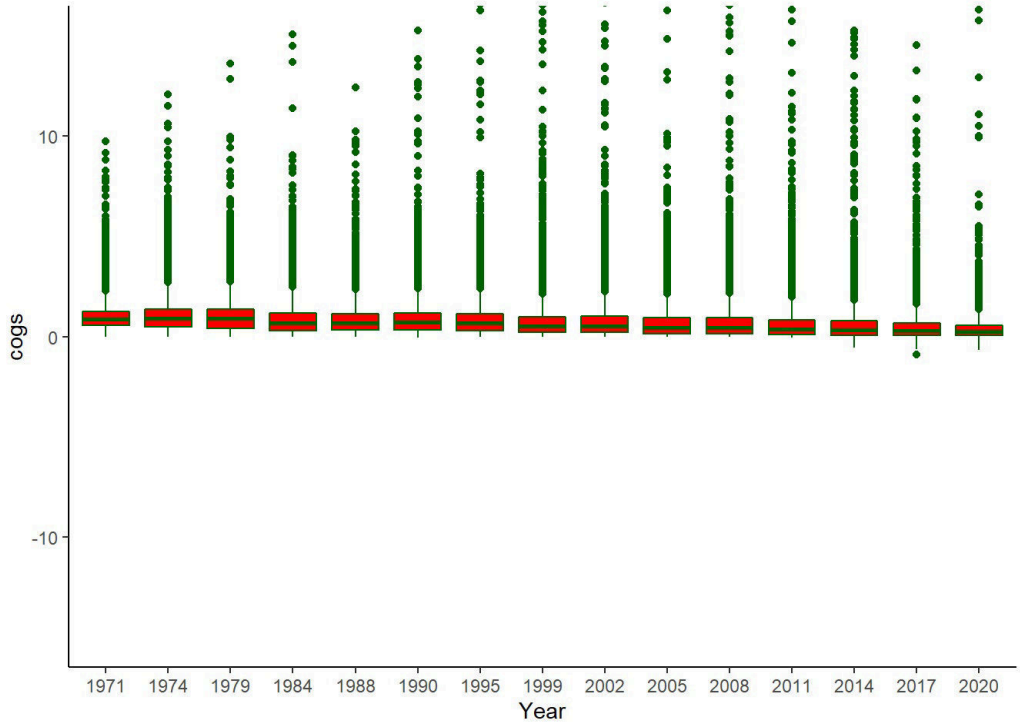

pstk
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

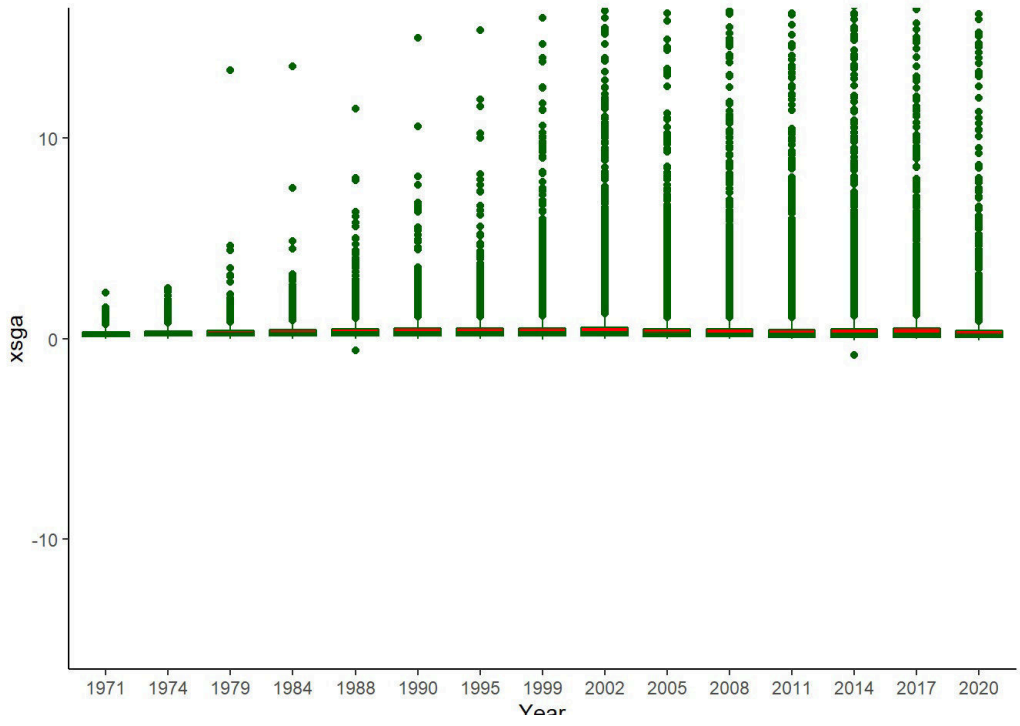


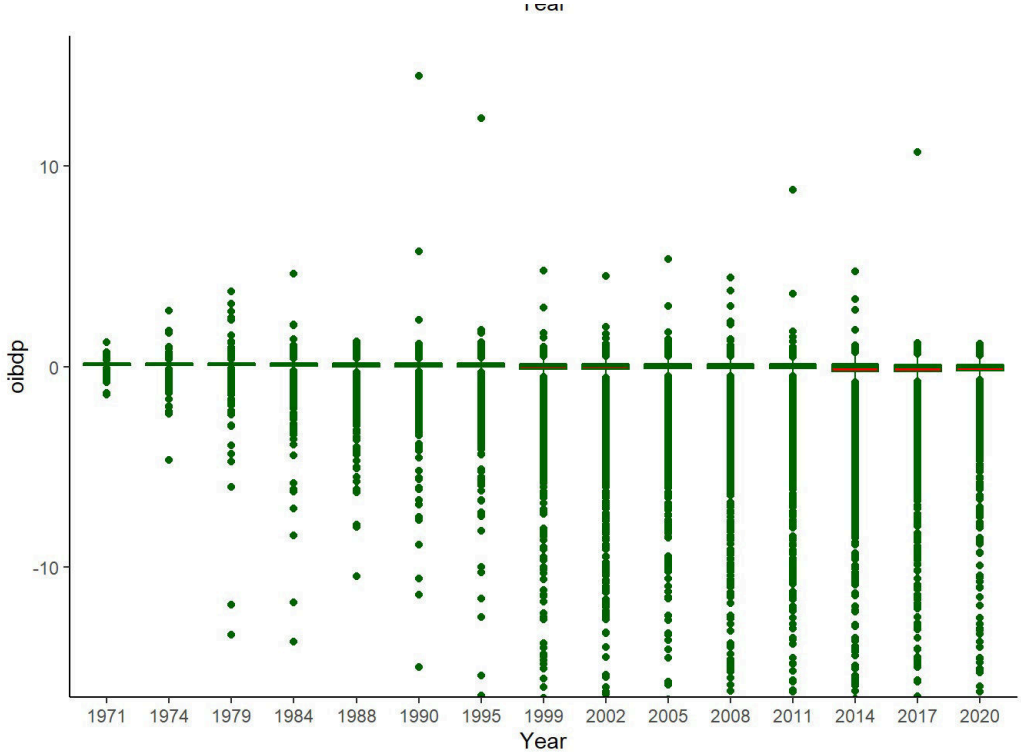


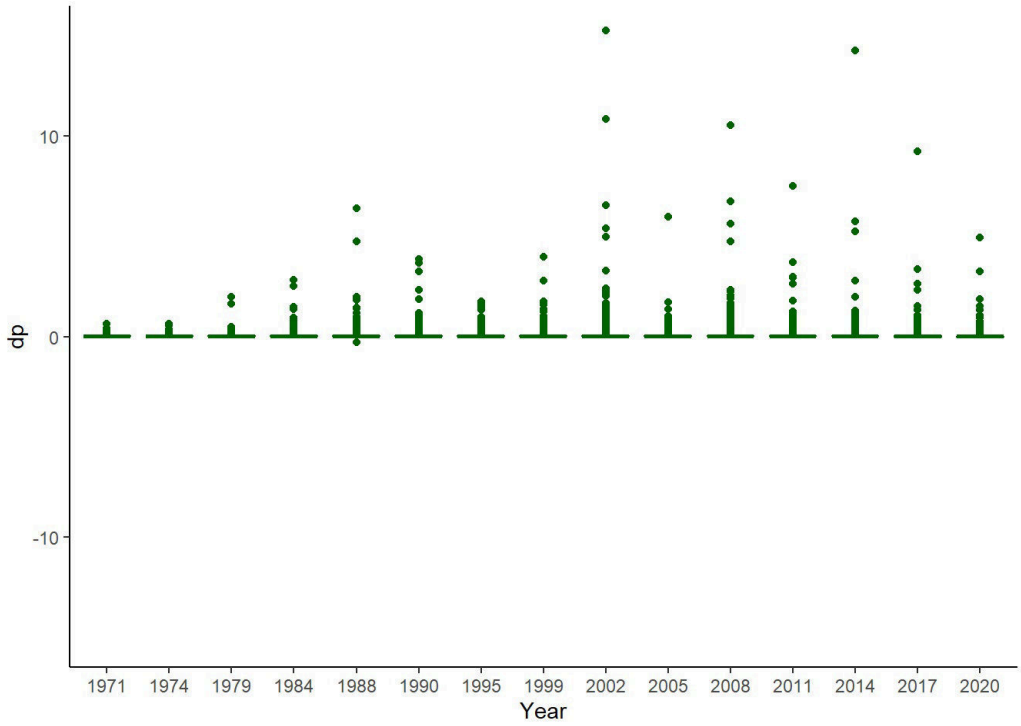

```
plot_all_boxplots(Income_Statement_Data)
```

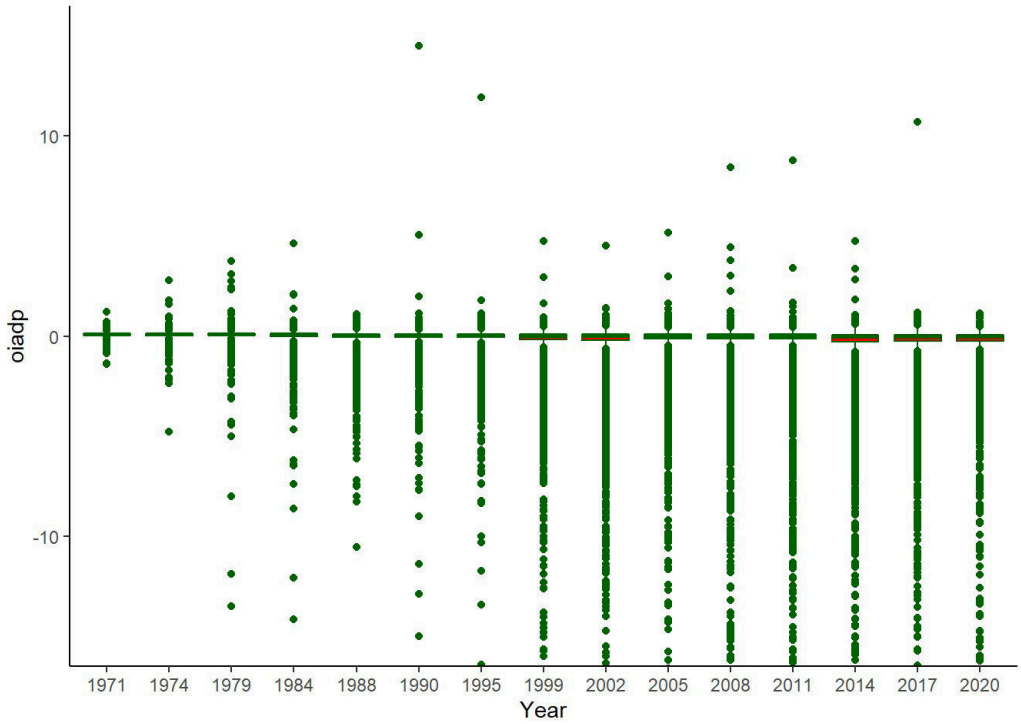

sale
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

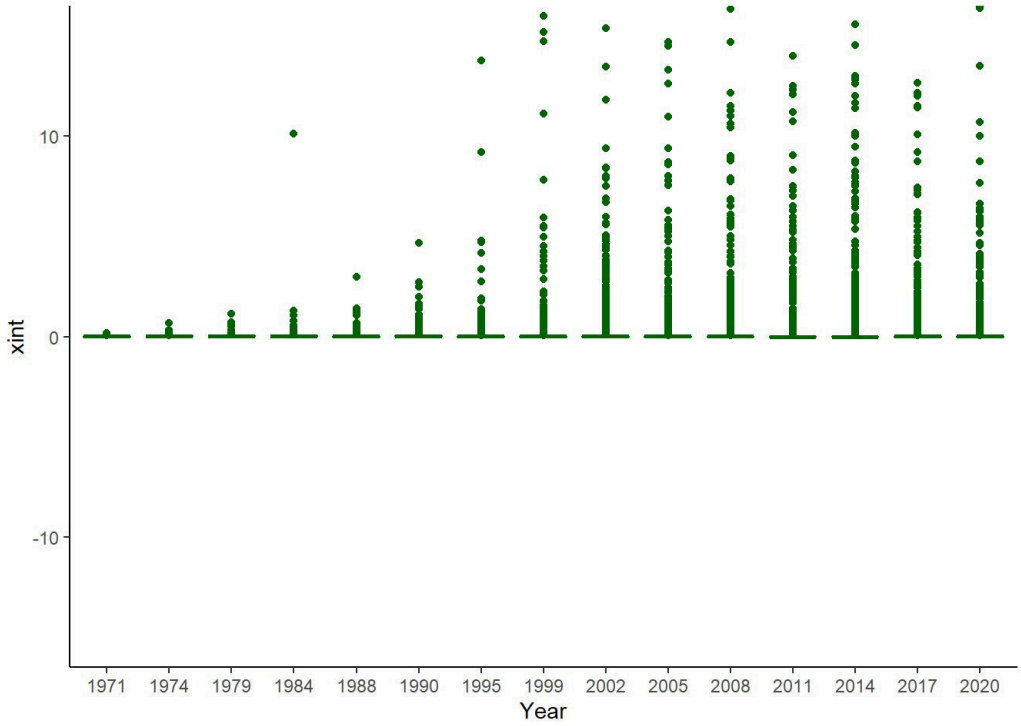

cogs
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

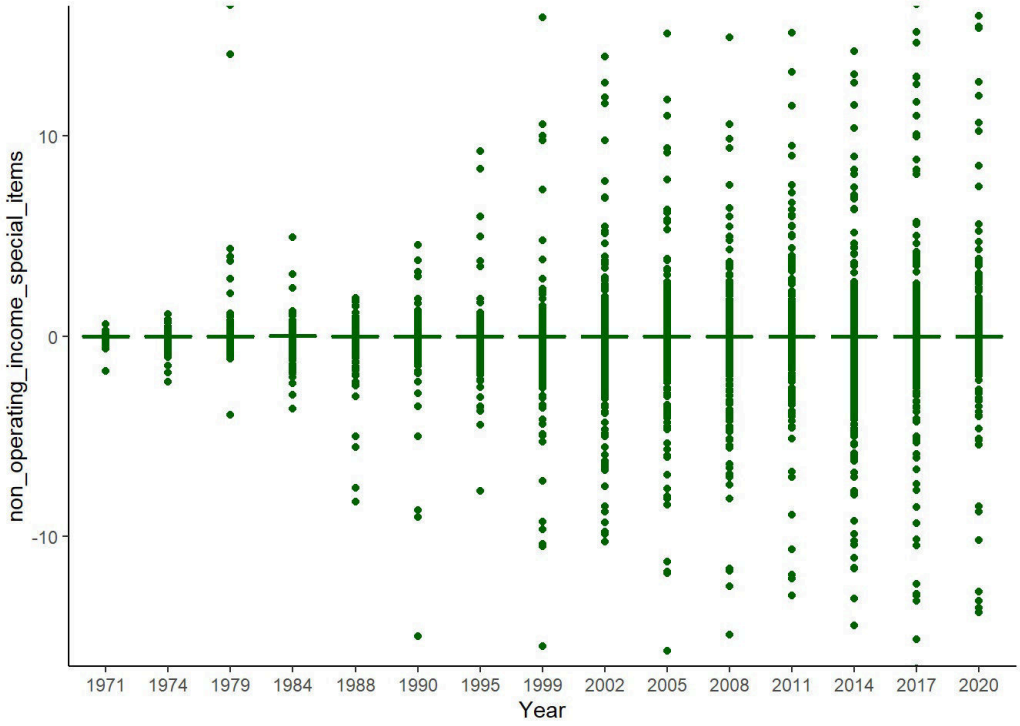

xsga
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

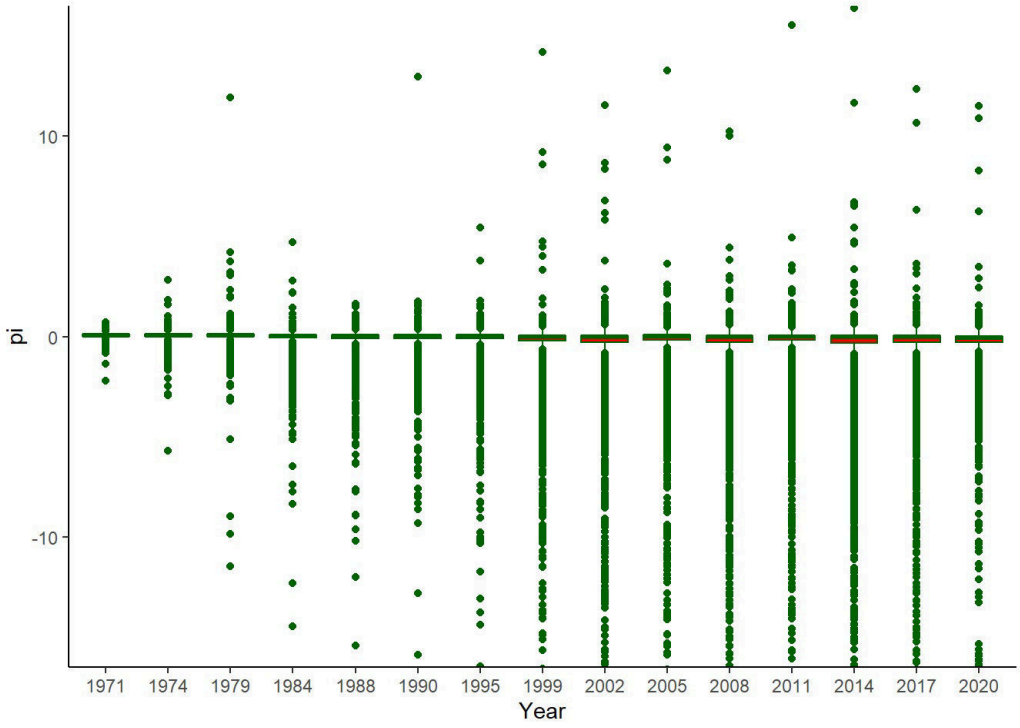

oibdp
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

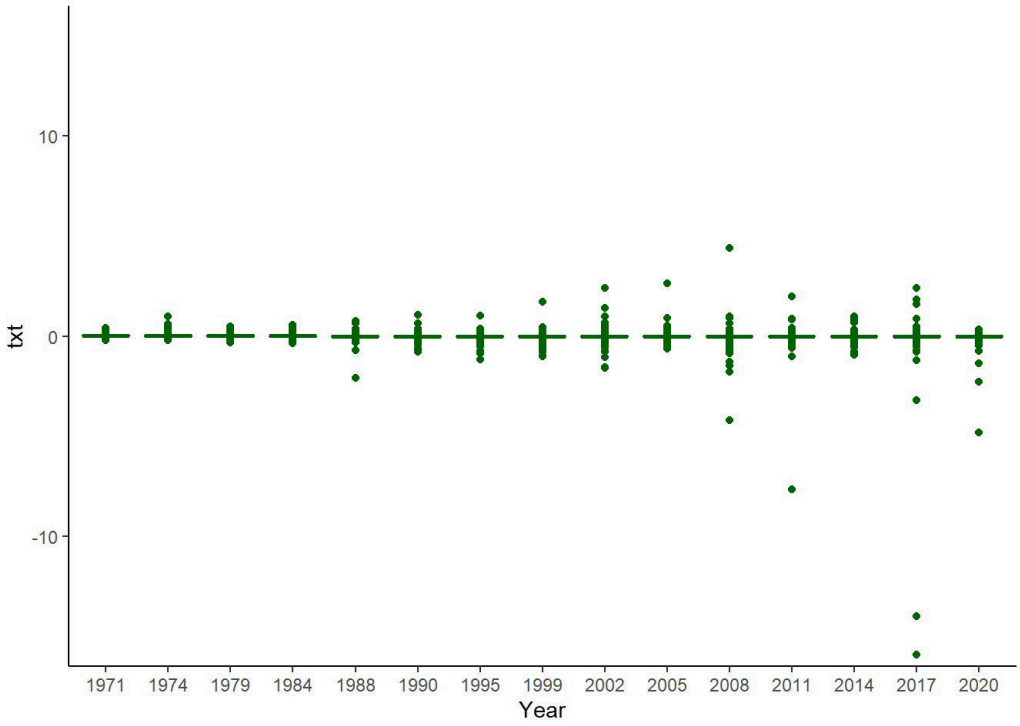

dp
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

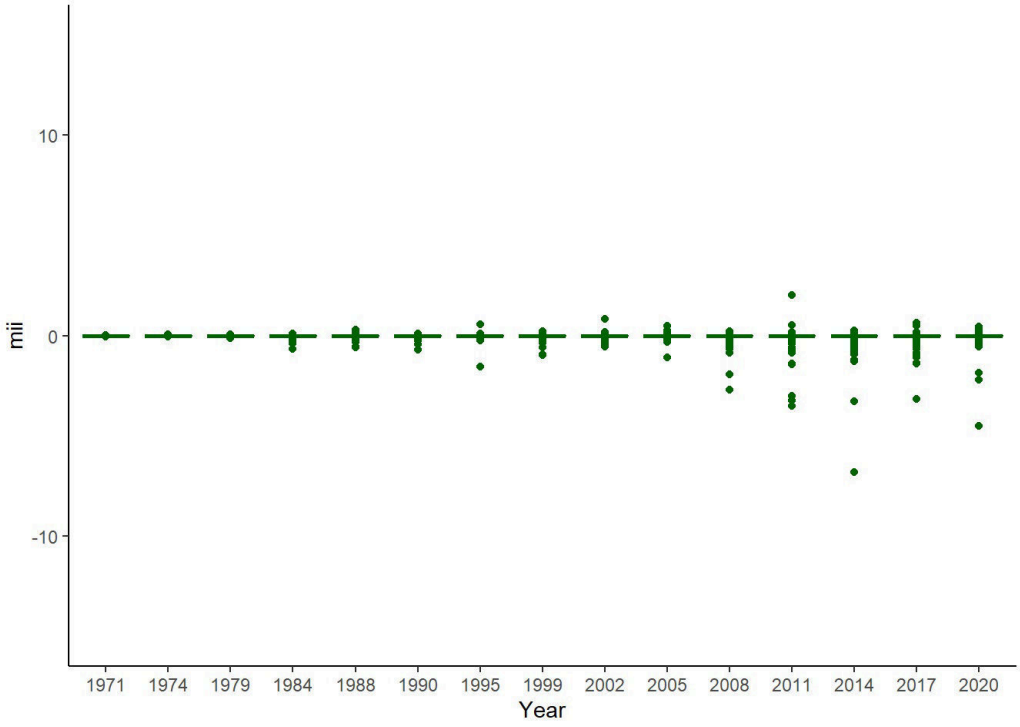

oiadp
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

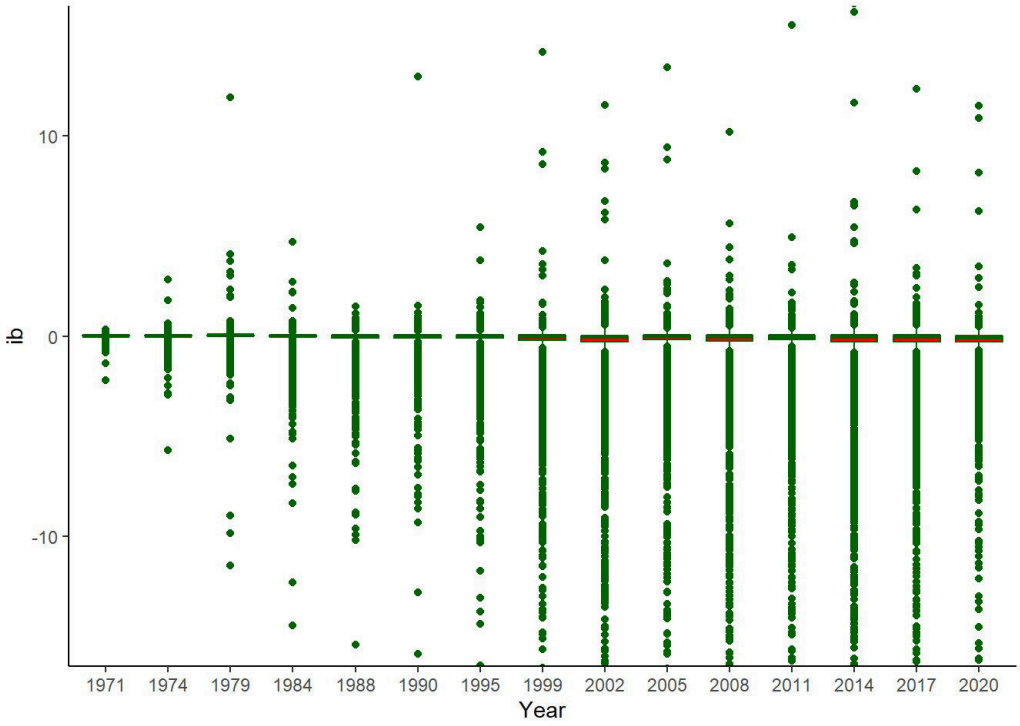

xint
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

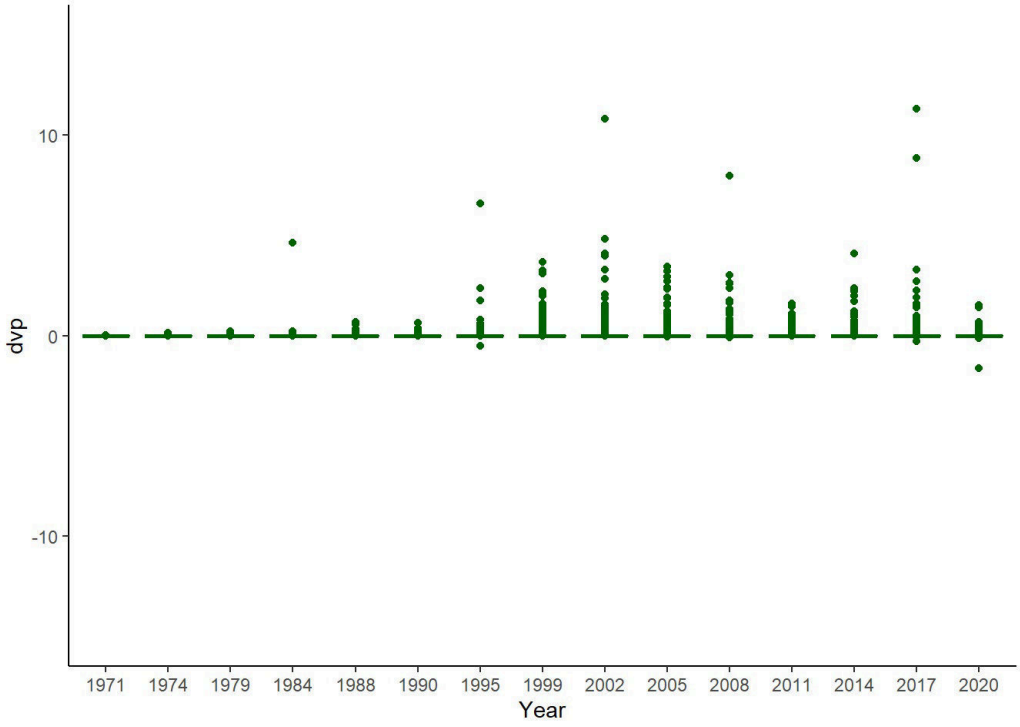

non_operating_income_special_items
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

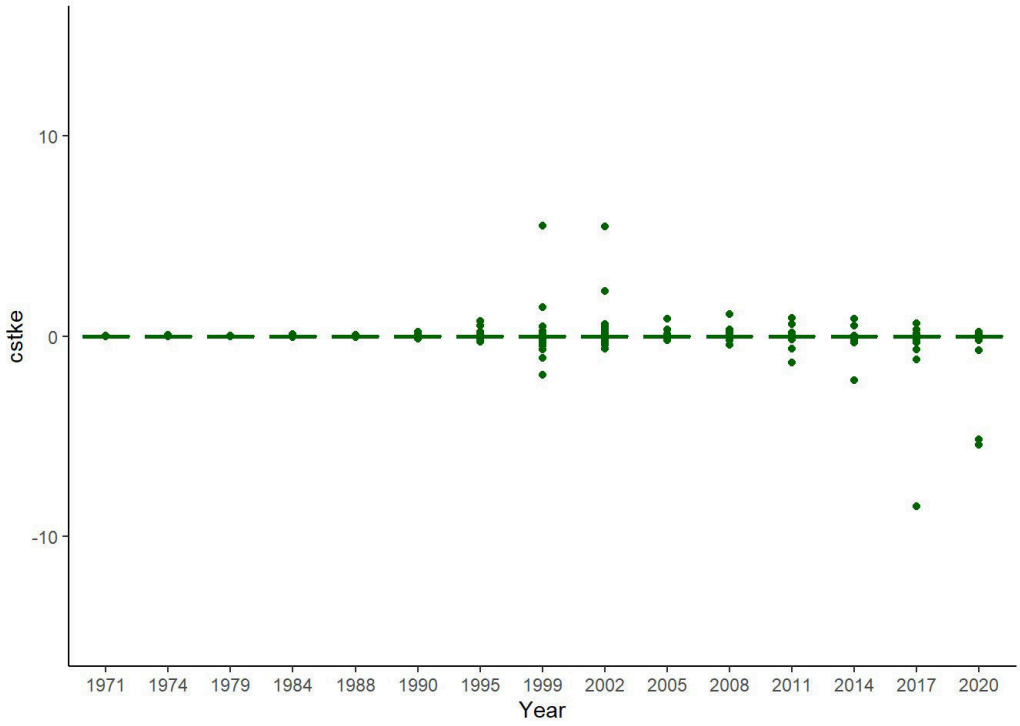

pi
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

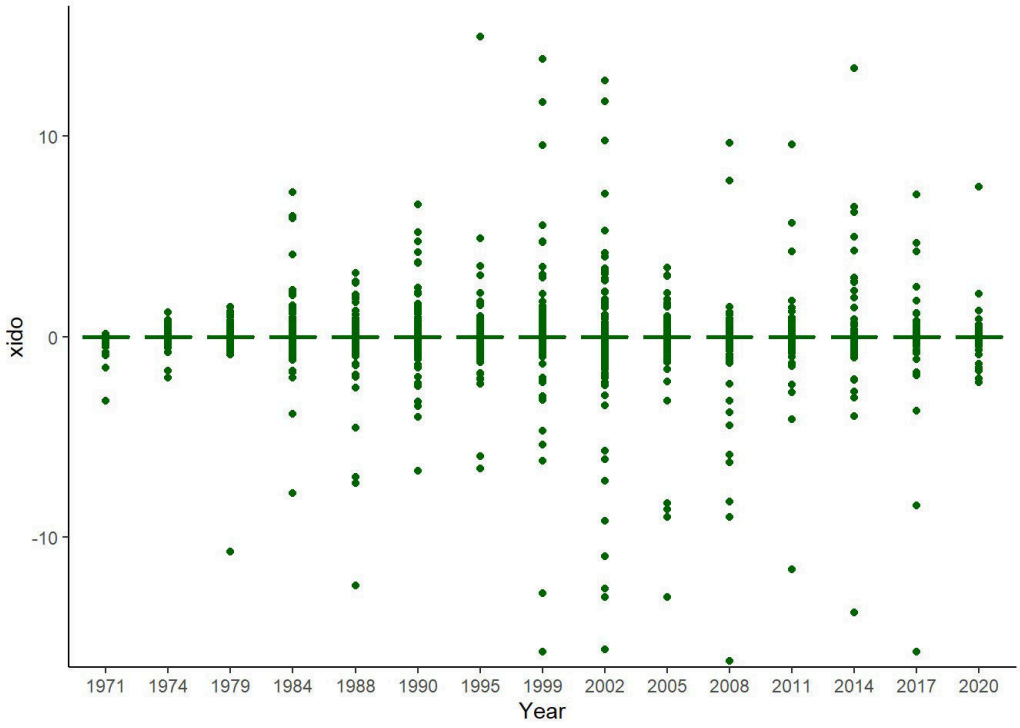

txt
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

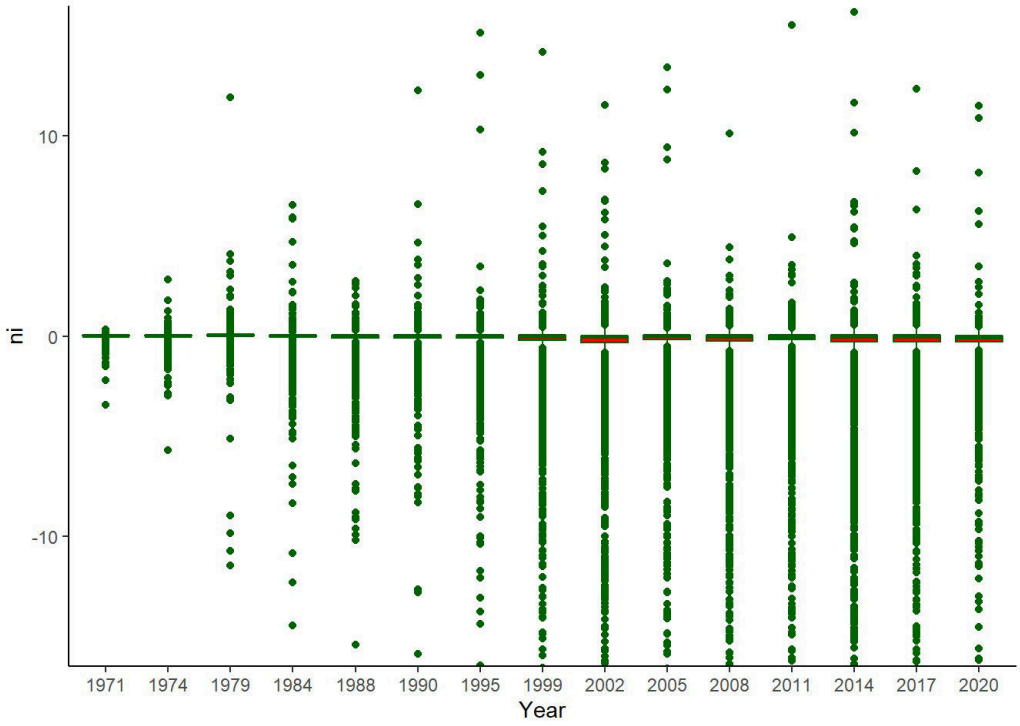

mii
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


ib
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

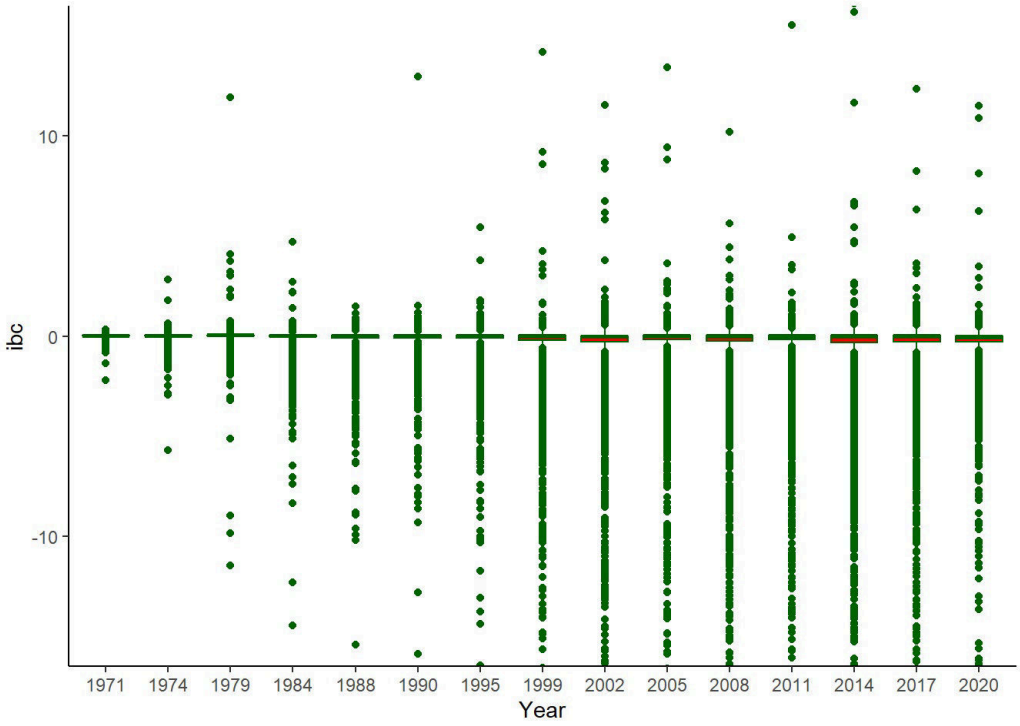

dvp
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

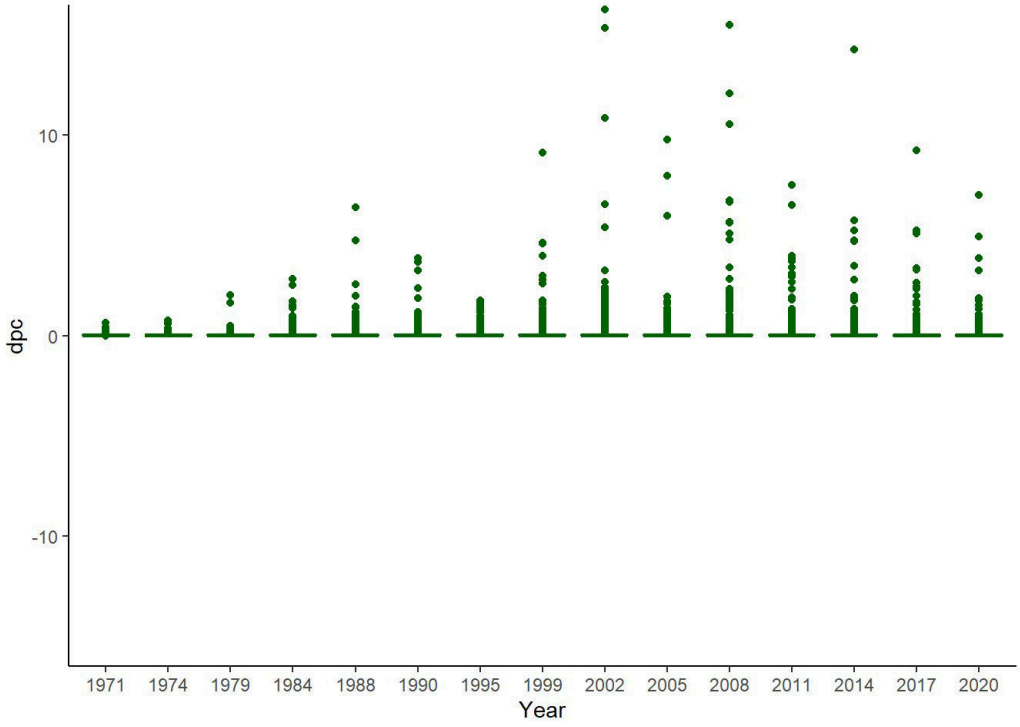

cstke
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

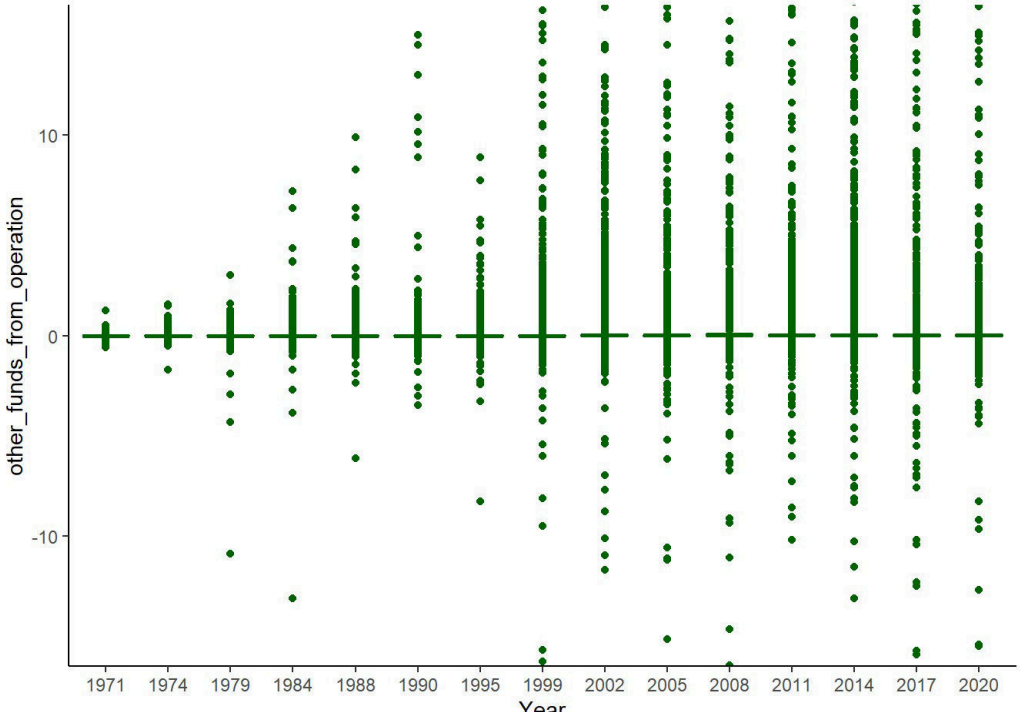

xido
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

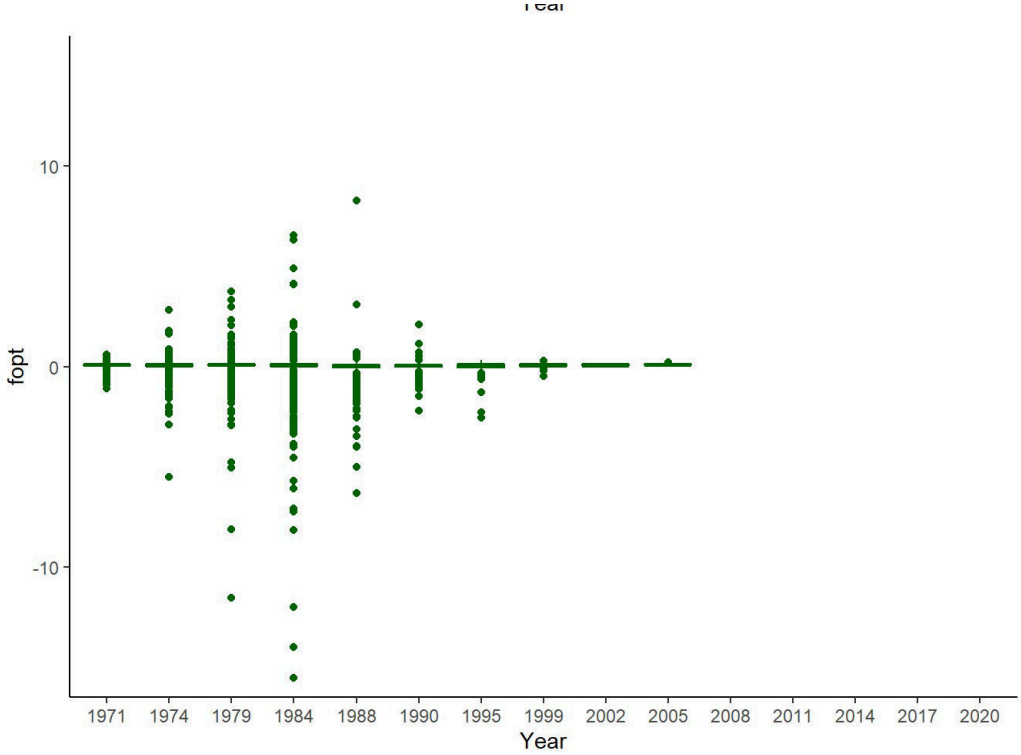


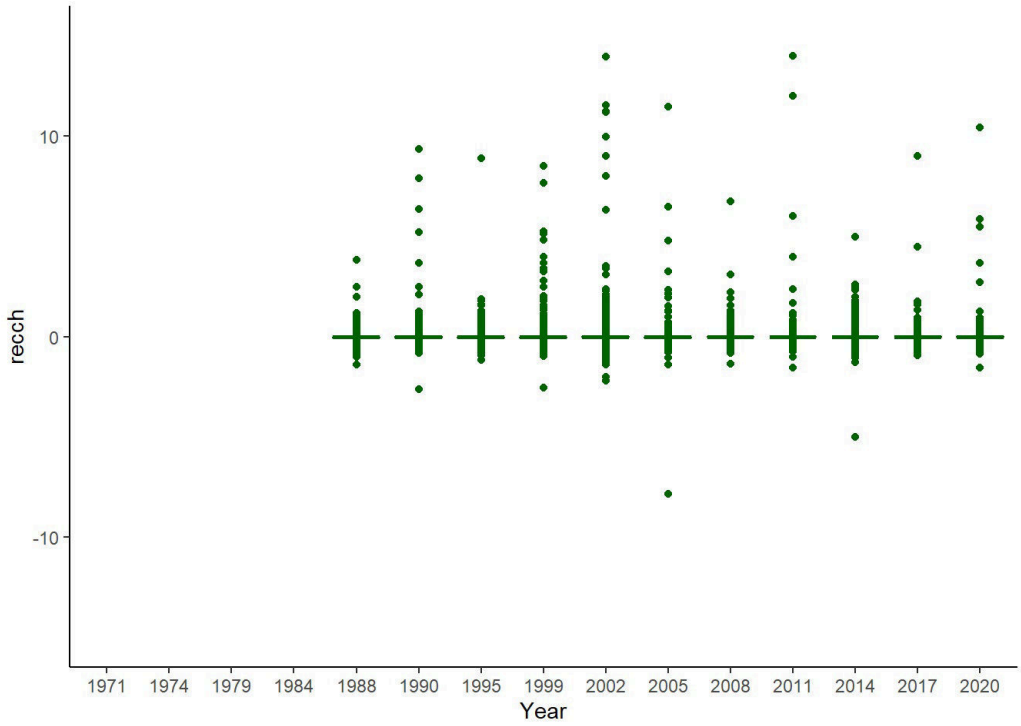

```
plot_all_boxplots(Cash_Flow_Data)
```

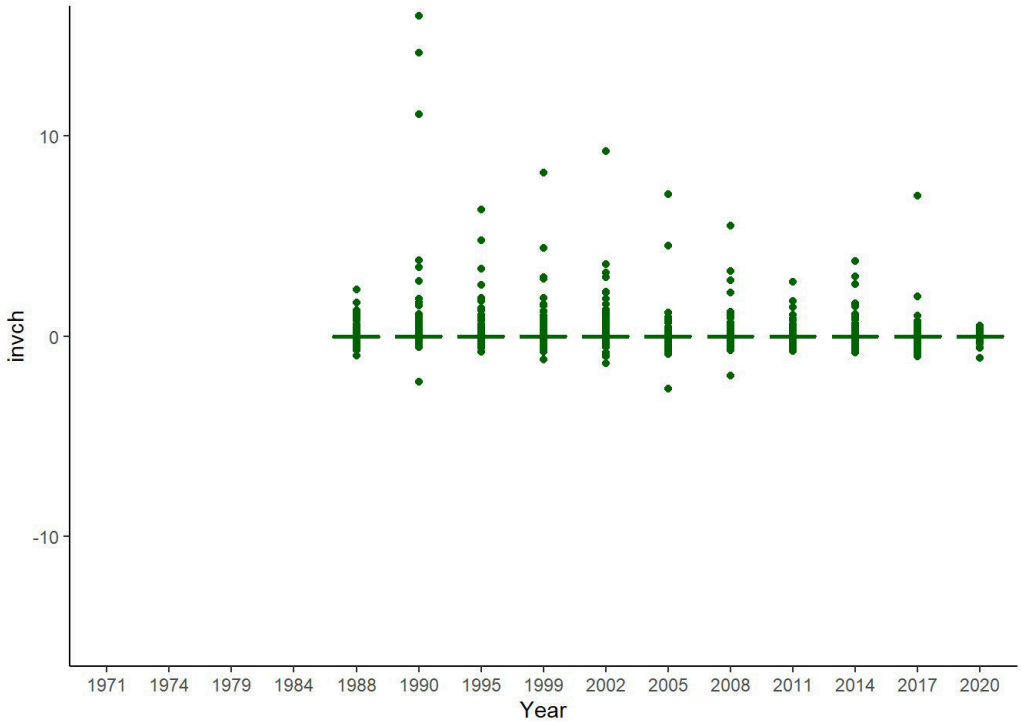

ibc
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

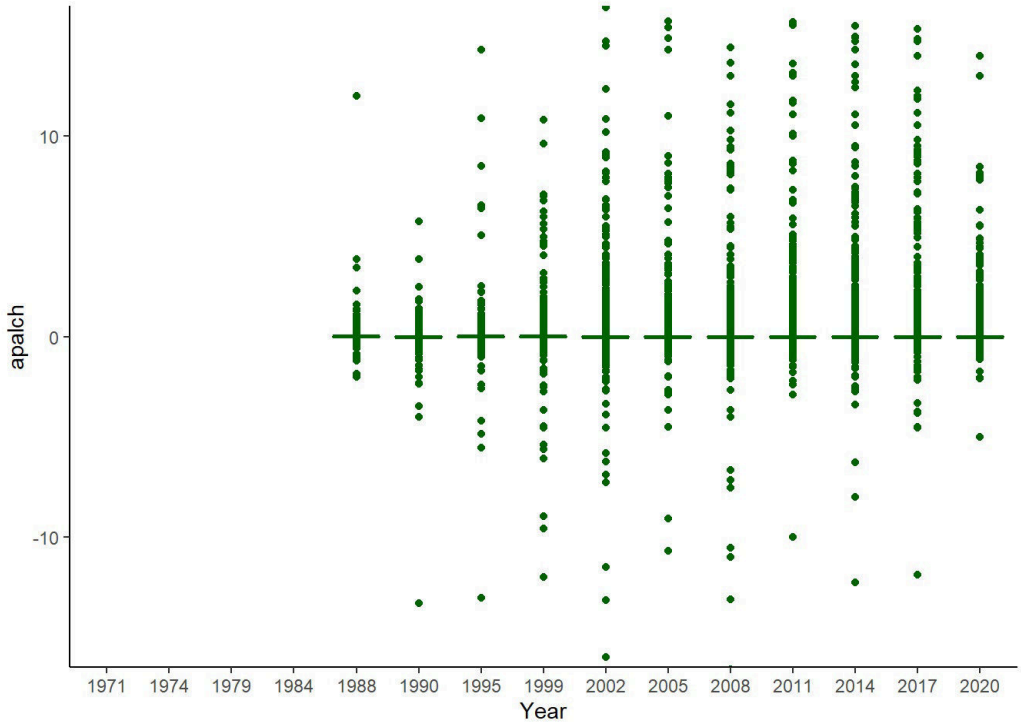

dpc
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

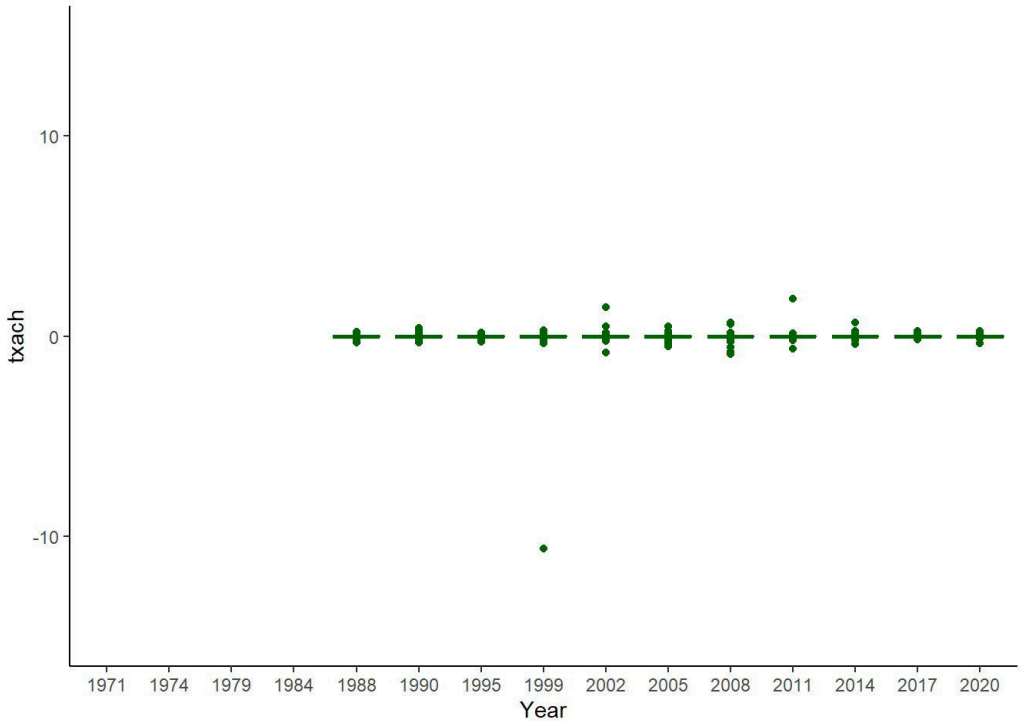

other_funds_from_operation
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

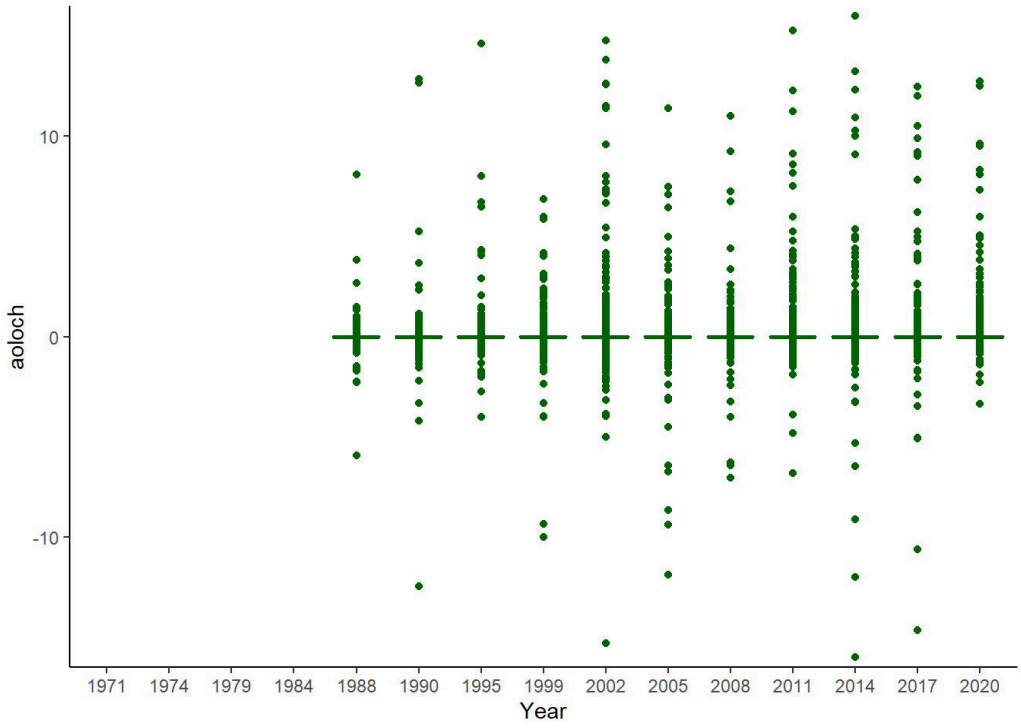

fopt
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

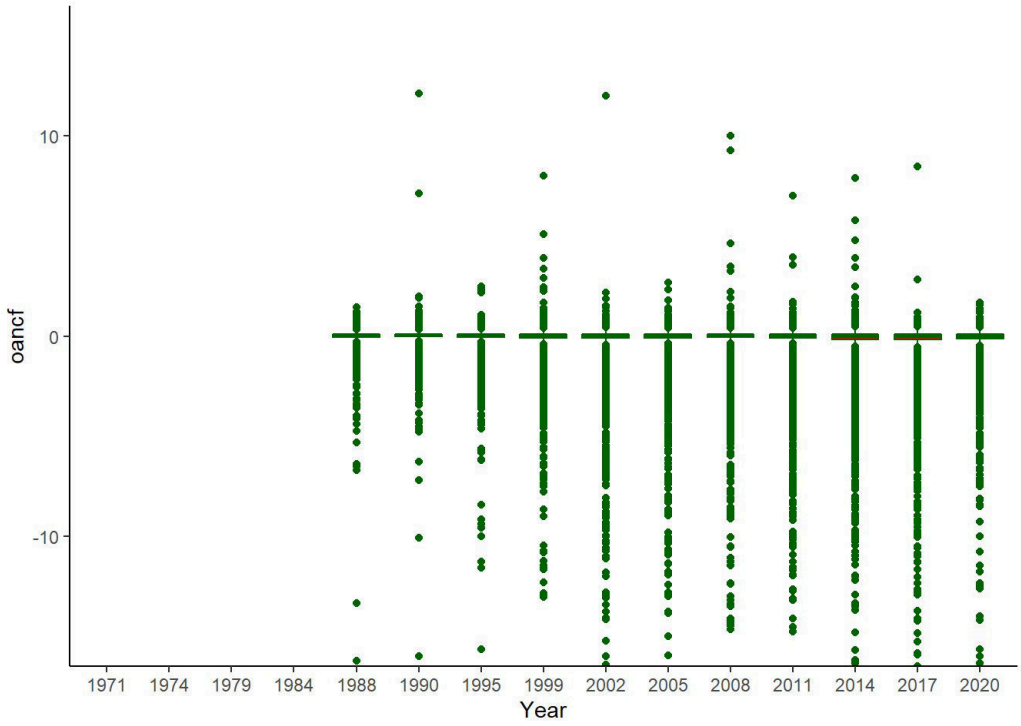

recch
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

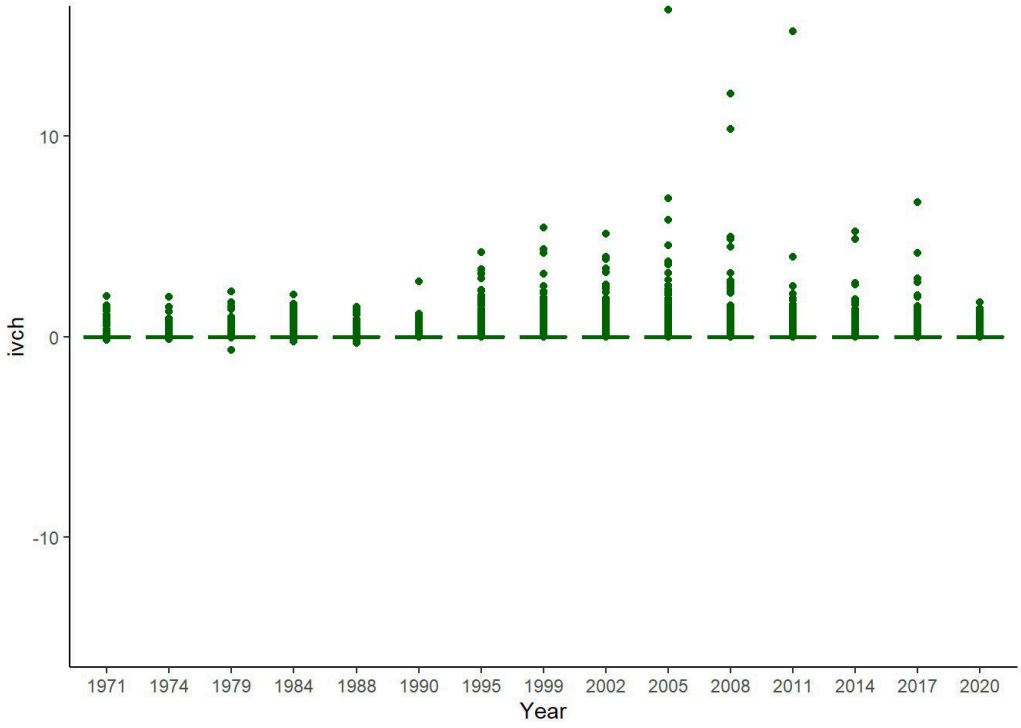

10
0
-10
invch
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

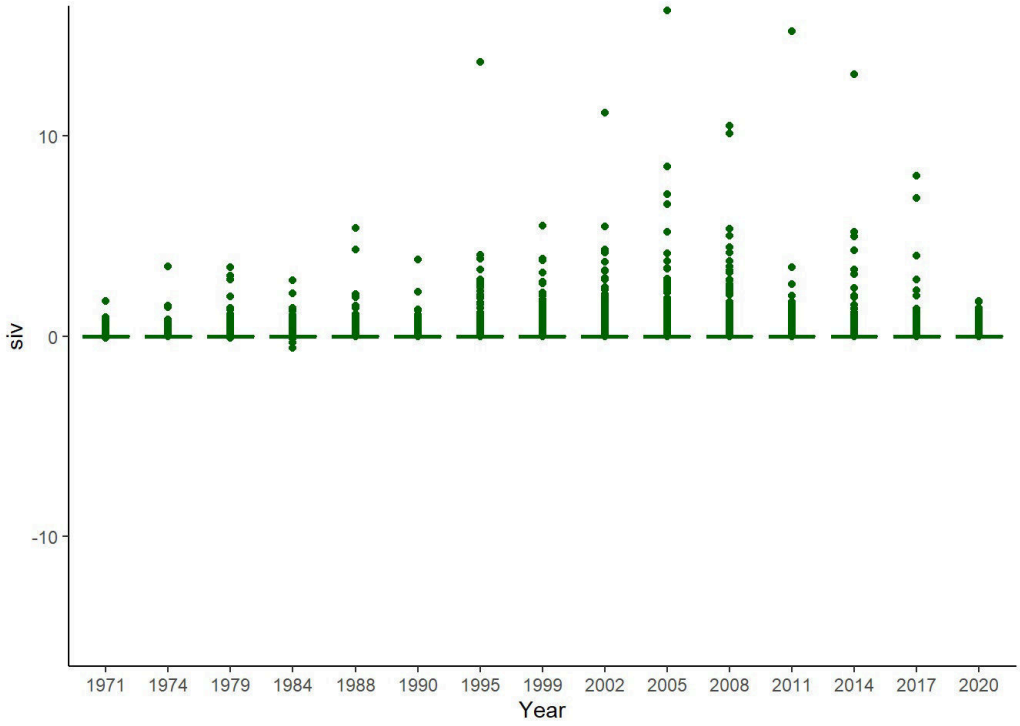

10
0
-10
apalch
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

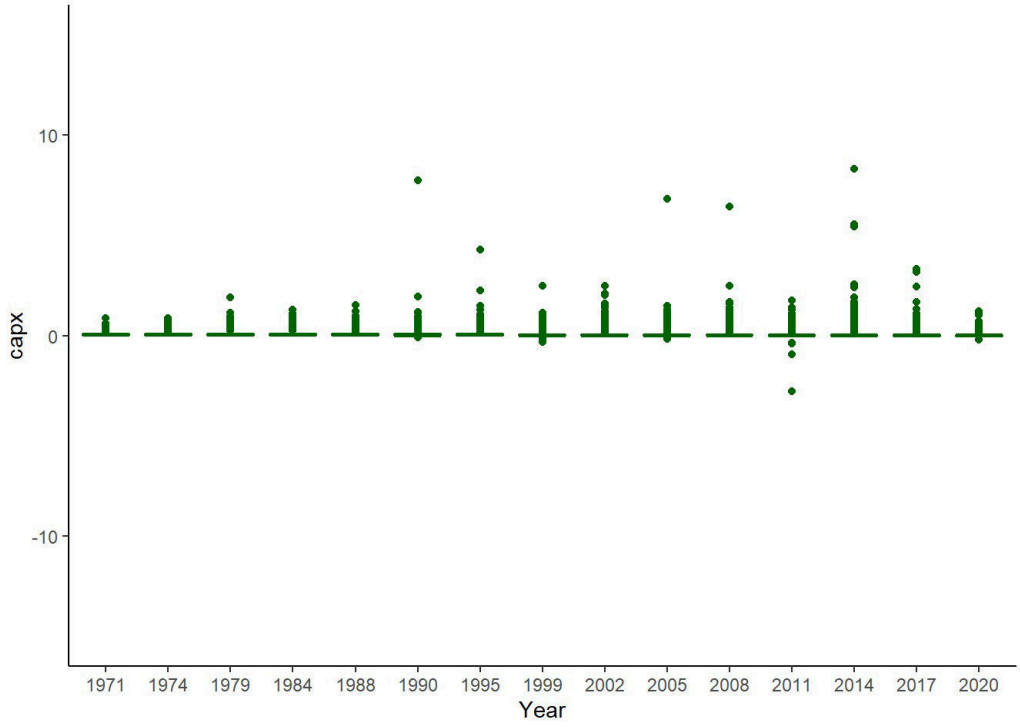

txach
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

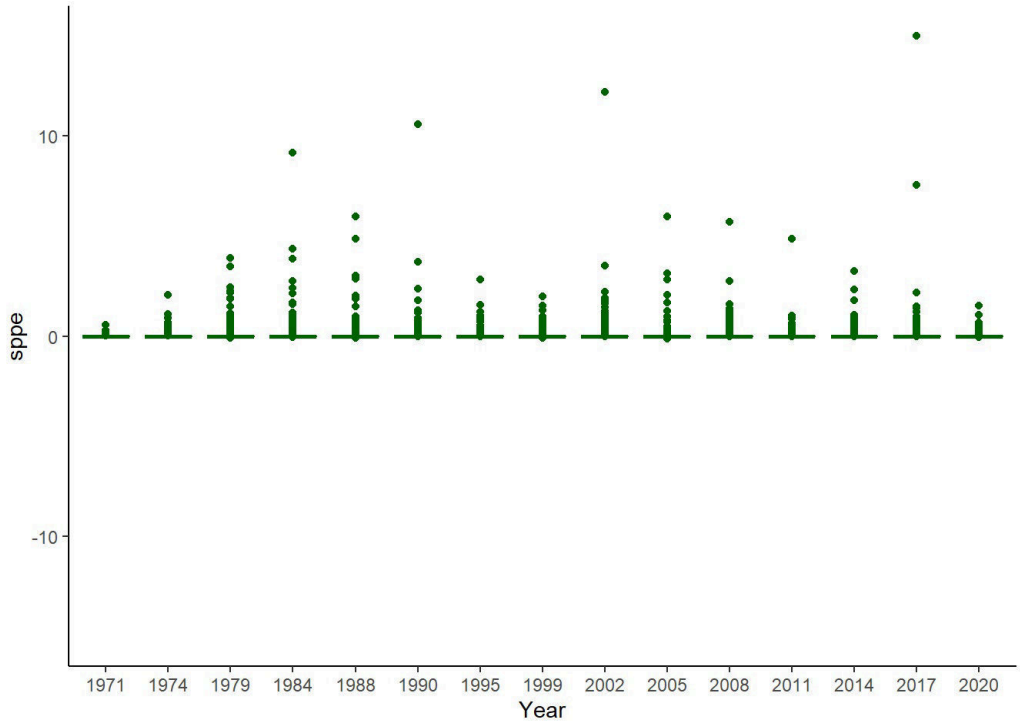

aoloch
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

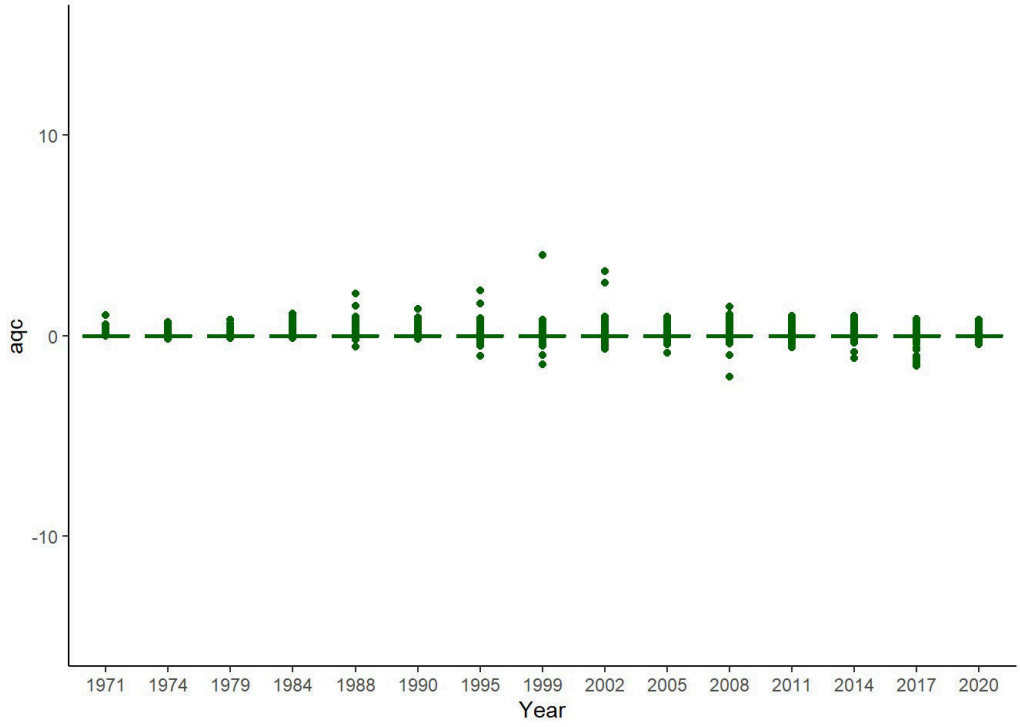

oancf
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

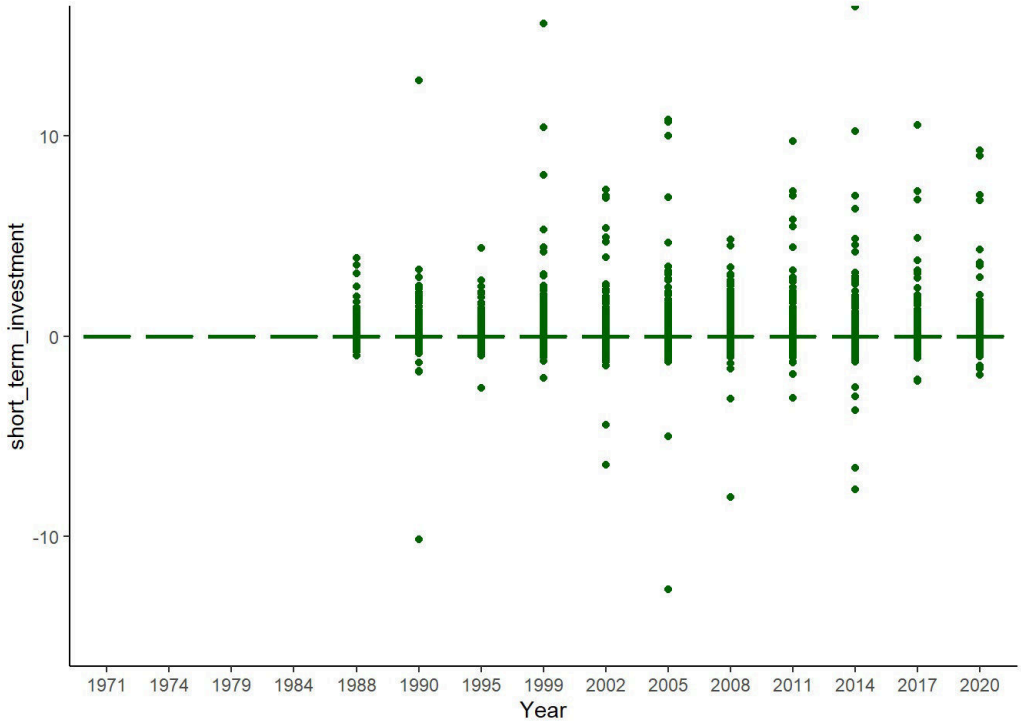

ivch
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

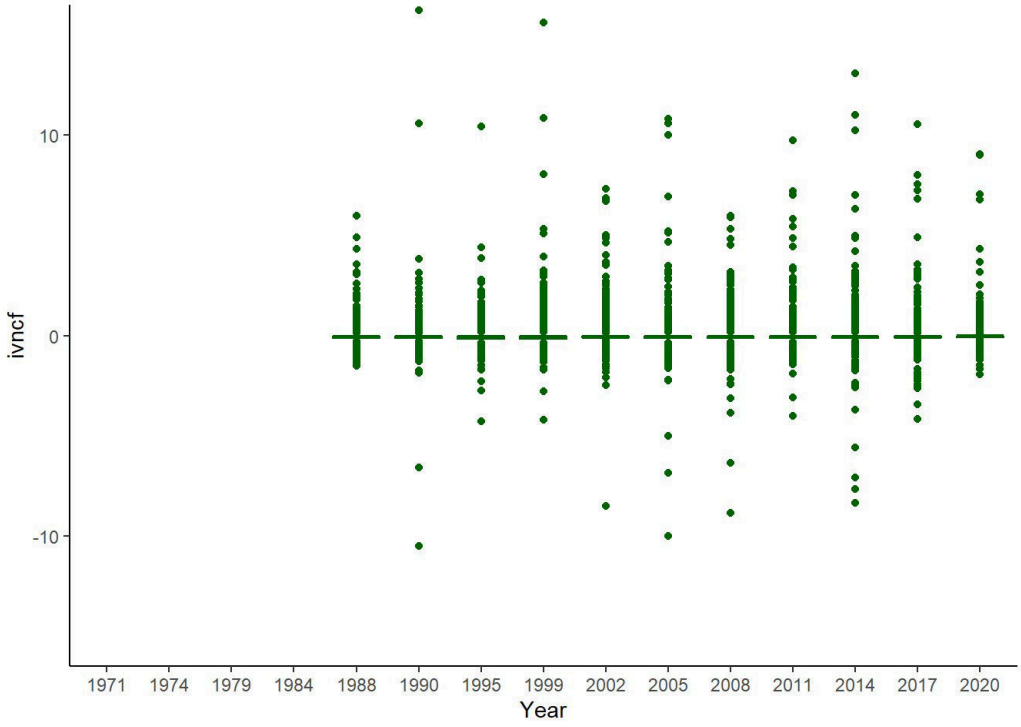

siv
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

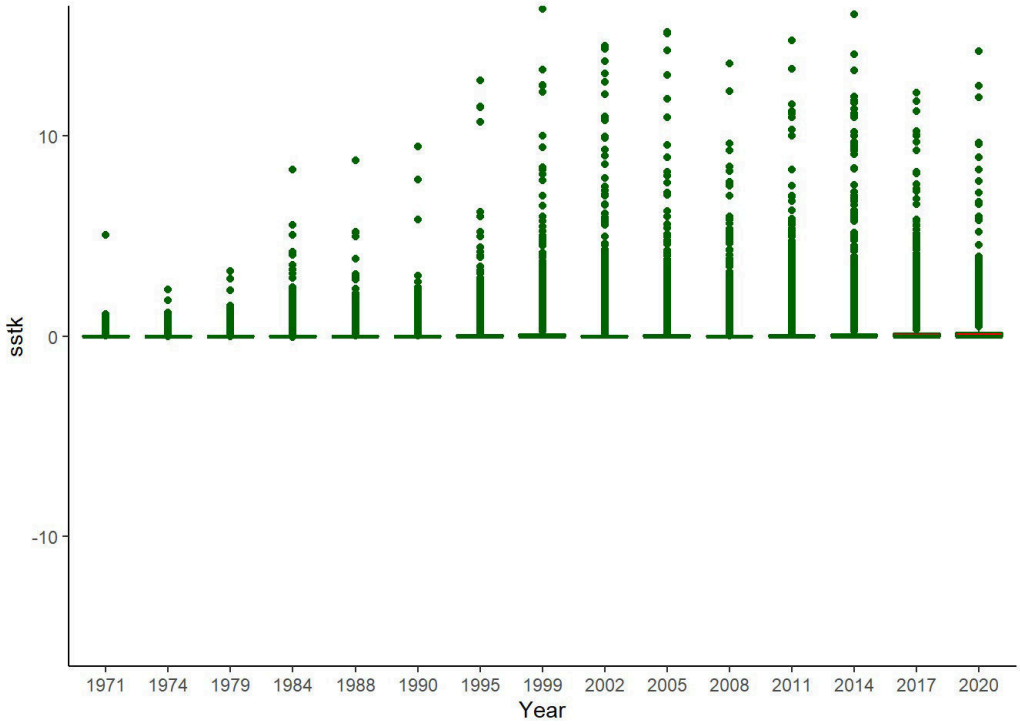

capx
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

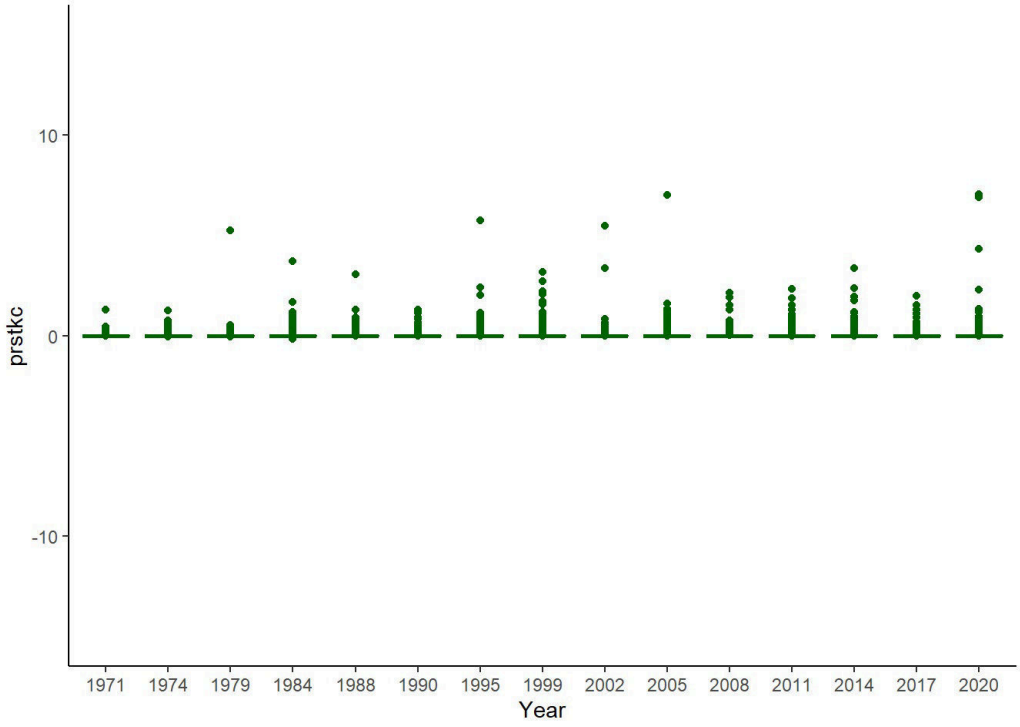

sppe
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

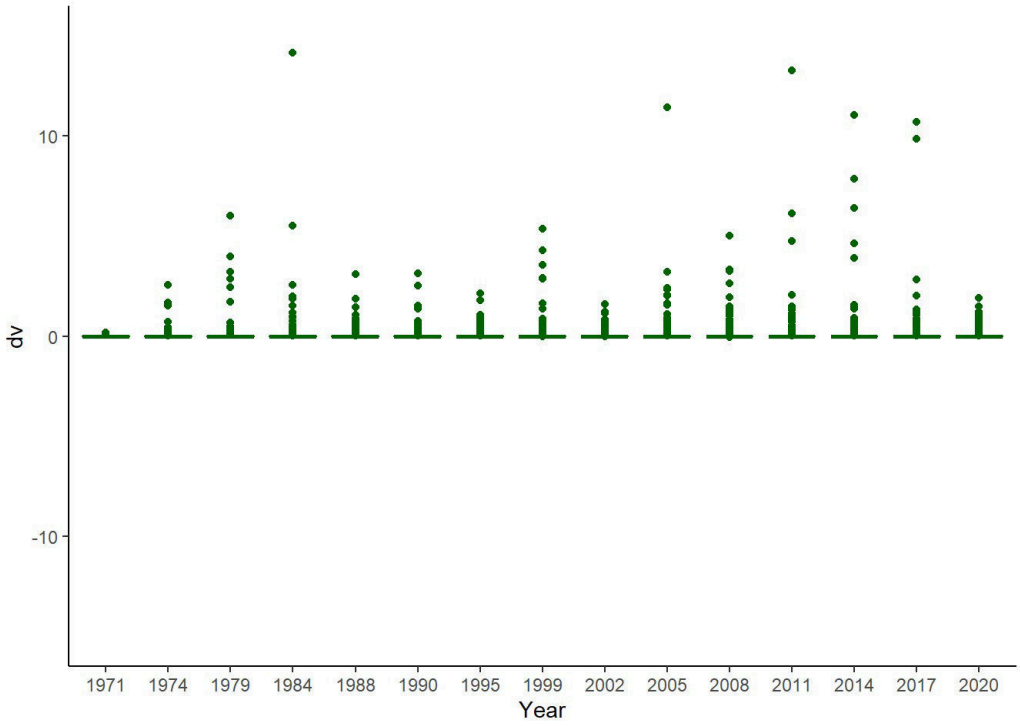

aqc
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

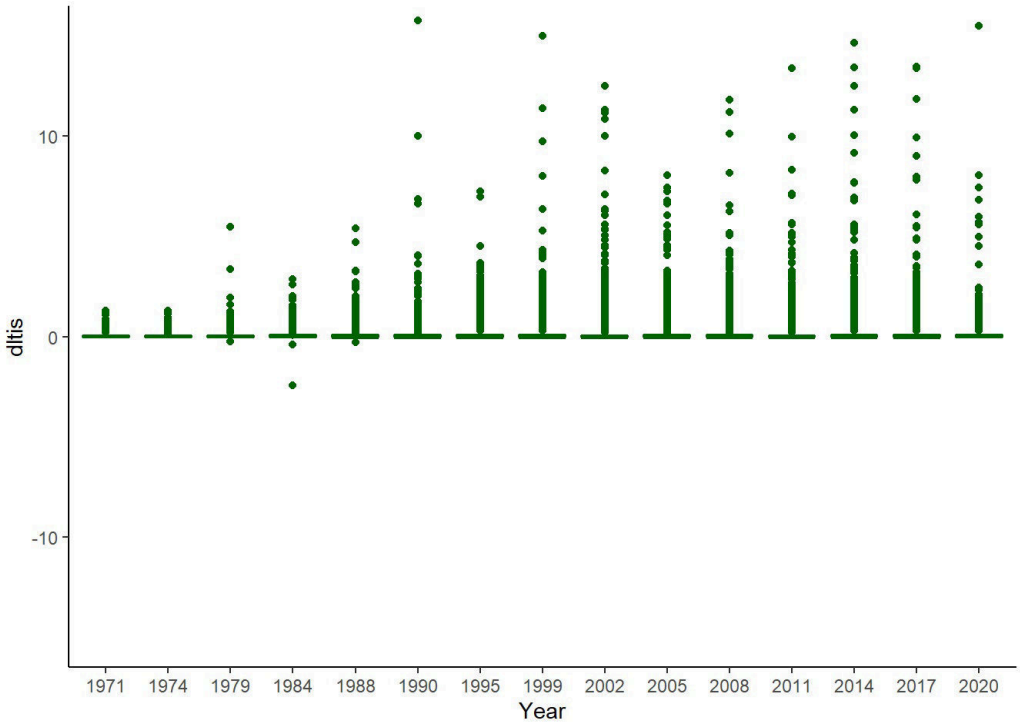

short_term_investment
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

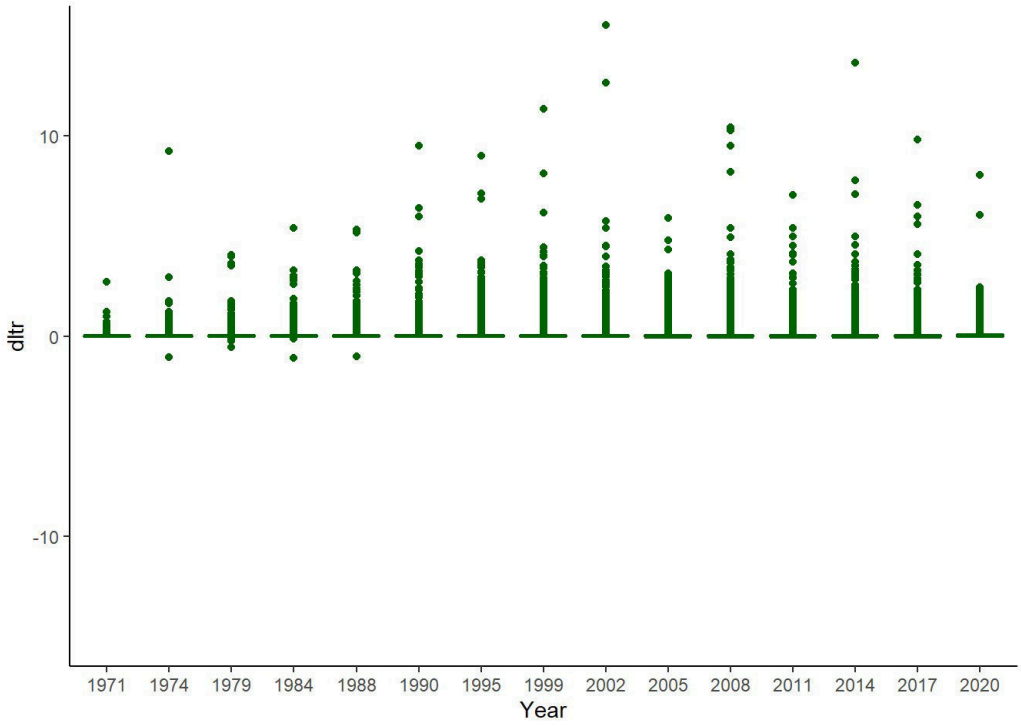

ivncf
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

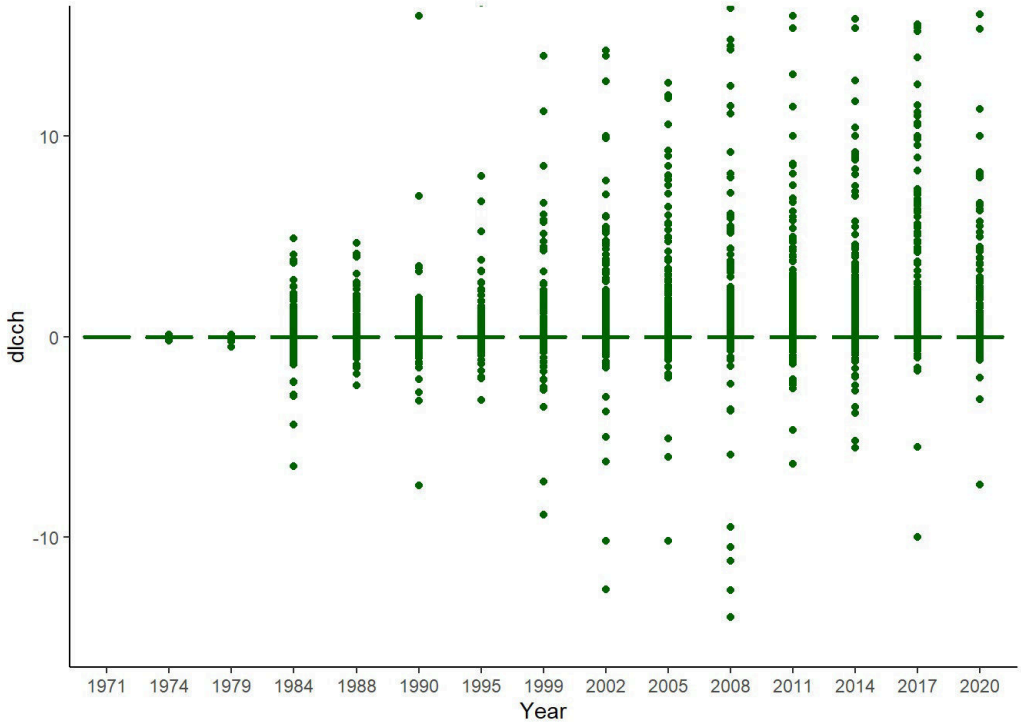

sstk
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

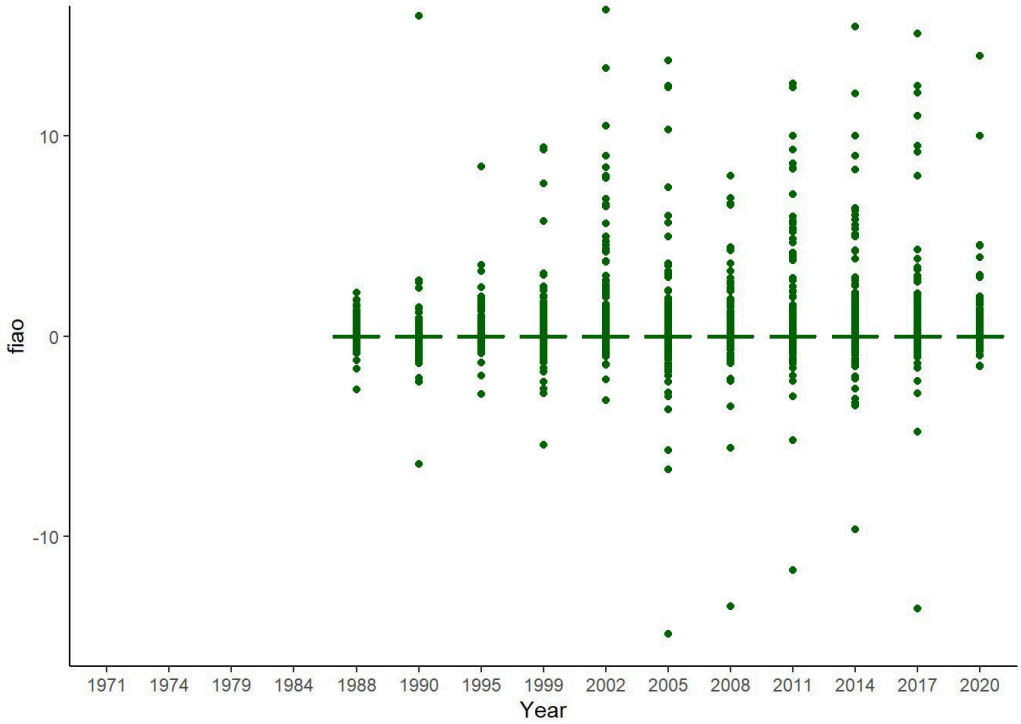

prstkc
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

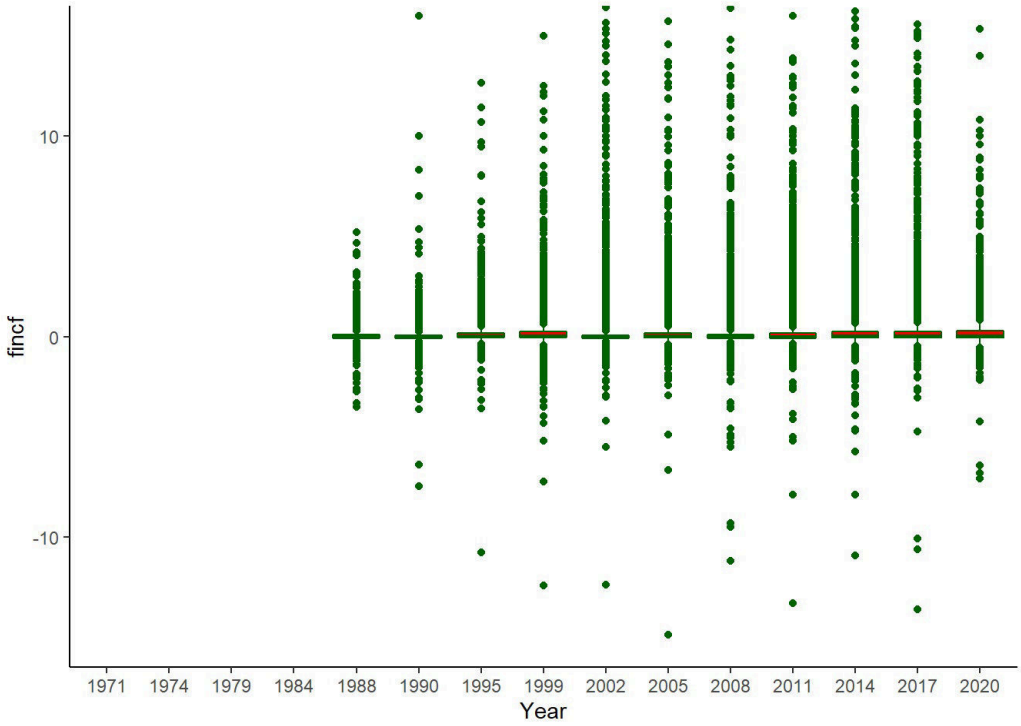

dv
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

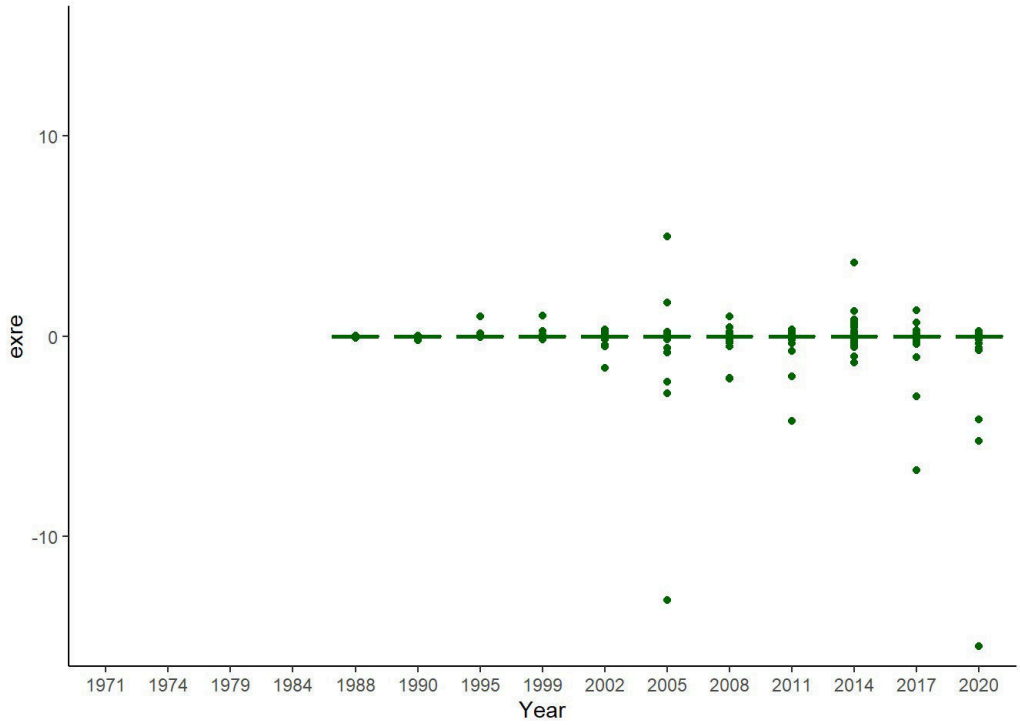

dltis
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

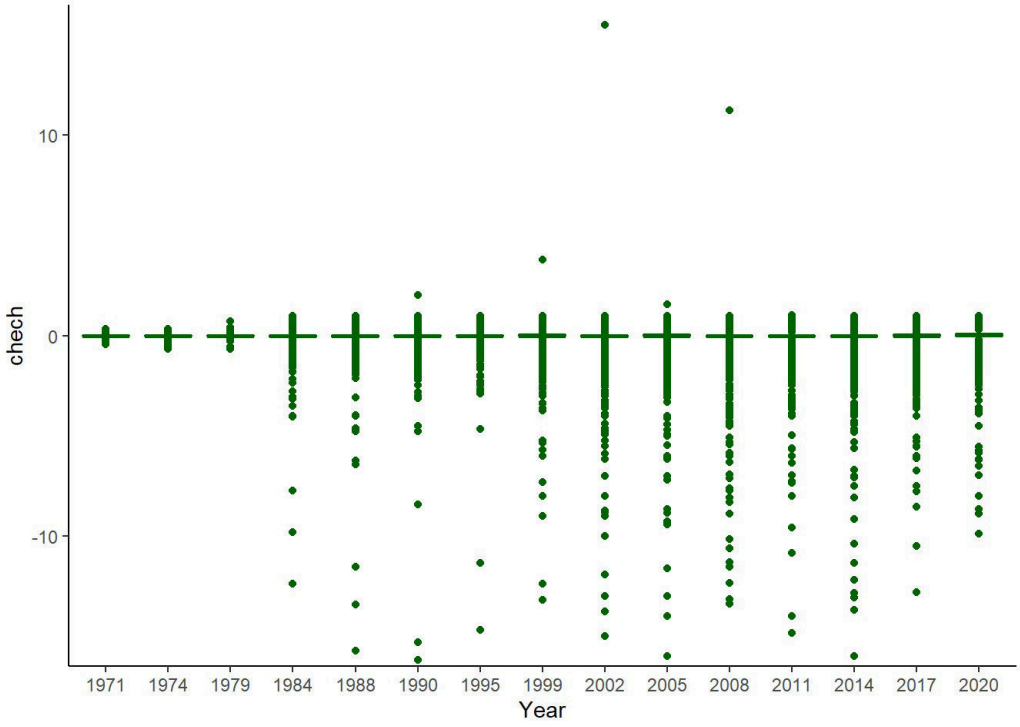

dltr
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

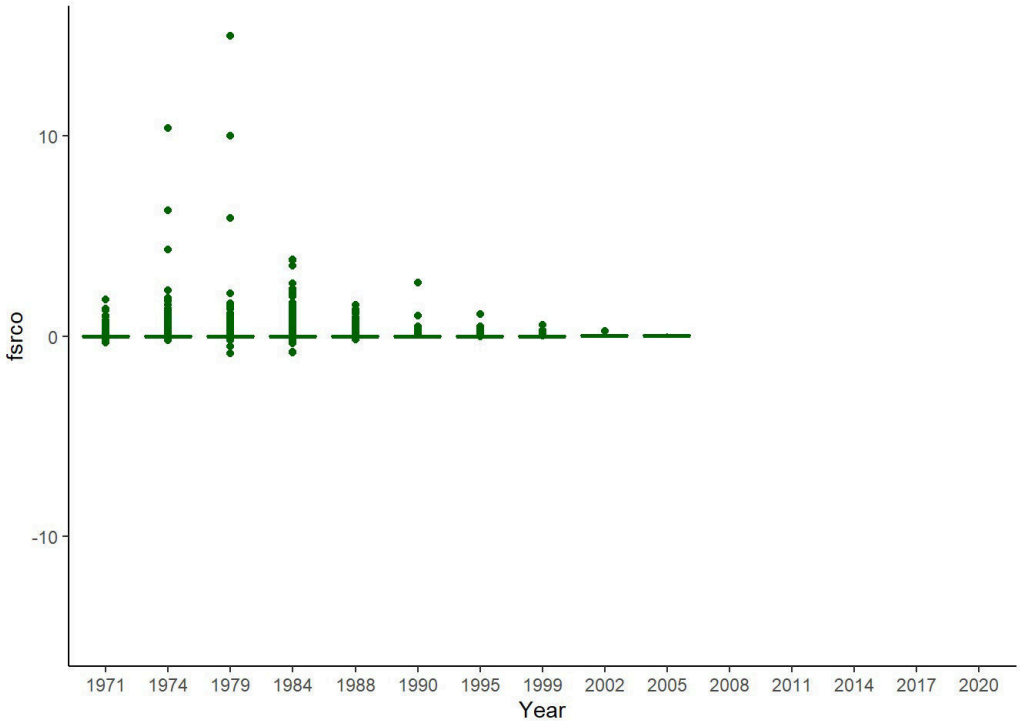

dlcch
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

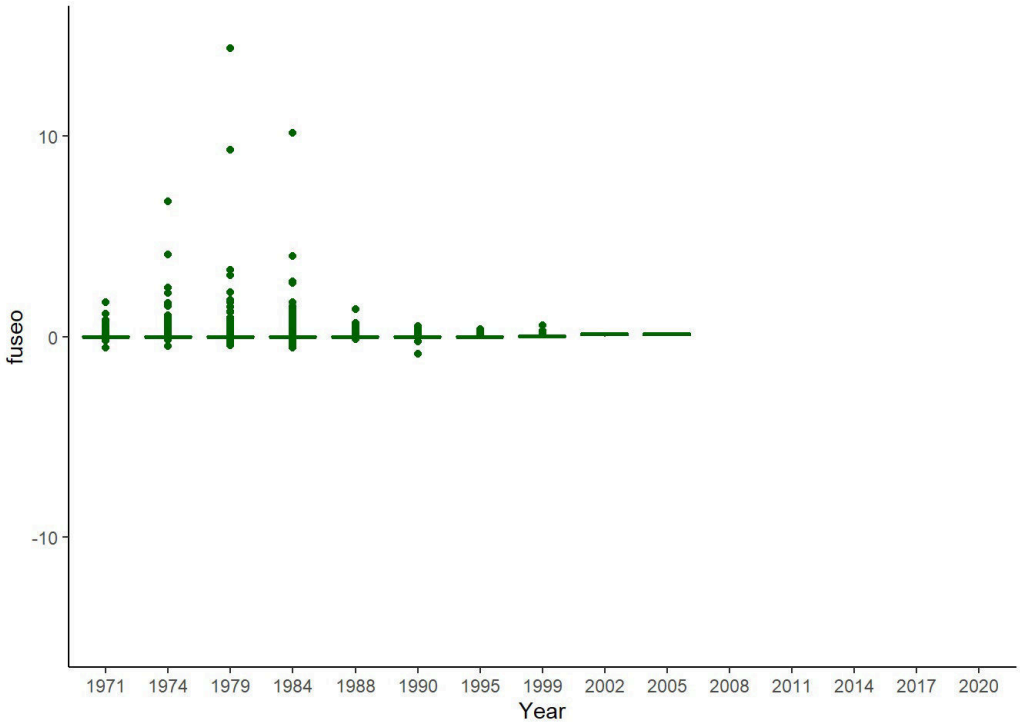

fiao
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

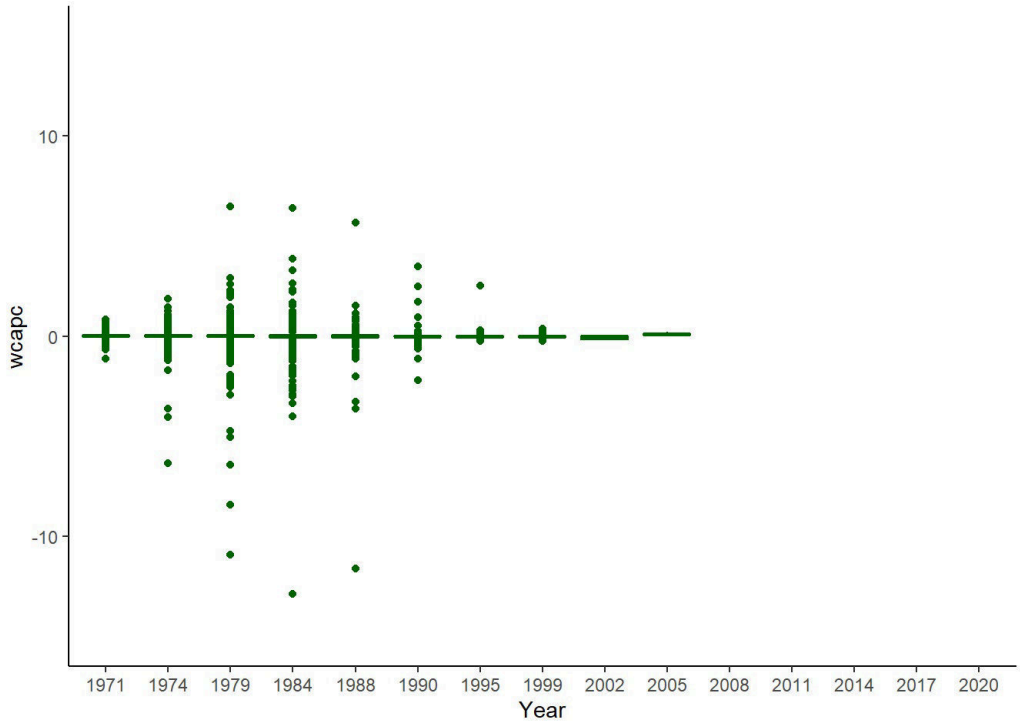

fincf
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


exre
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


chech
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

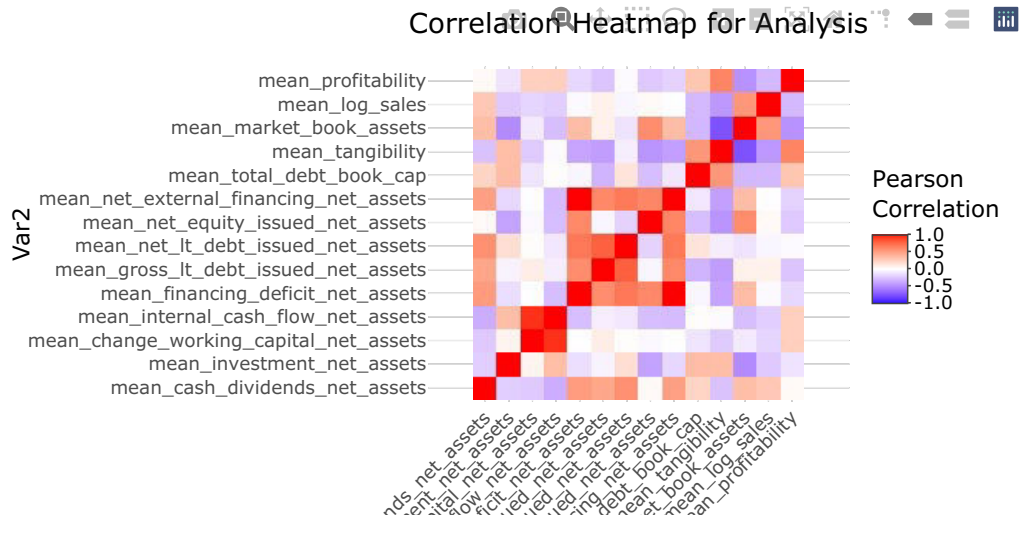

fsrco
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

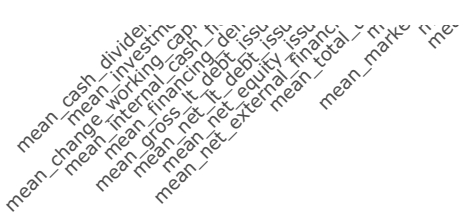

fuseo
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year



##Table 10(Correlation Features from Table 9)

```
#Adding new features to the Data
Modified_Data = Modified_Data %>%
  mutate(cash_dividends_net_assets =if_else((at-lct) !=0, dv/(at-lct),NA_real_),
         investment_net_assets = if_else((at-lct)!=0, if_else(scf > 0 & scf <= 3,(capx+ivch+aqc+fuseo-sppe-siv),i
f_else(scf == 7,(capx+ivch+aqc-sppe-siv-ivstch-ivaco),NA_real_))/(at-lct),NA_real_),
        change_working_capital_net_assets = if_else((at-lct)!=0,if_else(scf == 1,(wcapc+chech+dlcch),if_else(scf
== 2 & scf == 3,(-wcapc+chech-dlcch),if_else(scf == 7,(-recch-invch-apalch-txach-aoloch+chech-fiao-dlcch),NA_real
_)))/(at-lct),NA_real_),
        internal_cash_flow_net_assets = if_else((at-lct)!=0,if_else(scf > 0 & scf <= 3,(ibc+xidoc+dpc+txdc+ esubc
+sppiv+fopo+fsrco),if_else(scf == 7,(ibc+xidoc+dpc+txdc+esubc+sppiv+fopo+exre),NA_real_))/(at-lct),NA_real_),
        financing_deficit_net_assets = cash_dividends_net_assets+investment_net_assets+change_working_capital_net
_assets-internal_cash_flow_net_assets,
        gross_lt_debt_issued_net_assets = if_else((at-lct)!=0,dltis/(at-lct),NA_real_),
        net_lt_debt_issued_net_assets = if_else((at-lct)!=0,(dltis - dltr)/((at-lct)),NA_real_),
        net_equity_issued_net_assets = if_else((at-lct)!=0,(sstk - prstkc)/((at-lct)),NA_real_),
        net_external_financing_net_assets = if_else((at-lct)!=0,(net_lt_debt_issued_net_assets+net_equity_issued_
net_assets),NA_real_),
        total_debt_book_cap = if_else((at-lt)!=0,(dlc+dltt)/(at-lt),NA_real_),
        tangibility = if_else(at!= 0,ppent/at,NA_real_),
        market_book_assets = mkvalt/at,
        log_sales = if_else(sale != 0,log(sale),NA_real_),
        profitability = ni/at)

#Descriptive stats for key variables
c(count_key_variables_t9,mean_key_variables_t9,p25_key_variables_t9,p50_key_variables_t9, p75_key_variables_t9, s
td_key_variables_t9,max_key_variables_t9,min_key_variables_t9)%<-%stats(Modified_Data[c(1,90:length(colnames(Modi
fied_Data)))])
#Heatmap for the above mean key variables over time
get_heatmap(mean_key_variables_t9,colnames(mean_key_variables_t9)[2:length(colnames(mean_key_variables_t9))],"Ana
lysis")
```





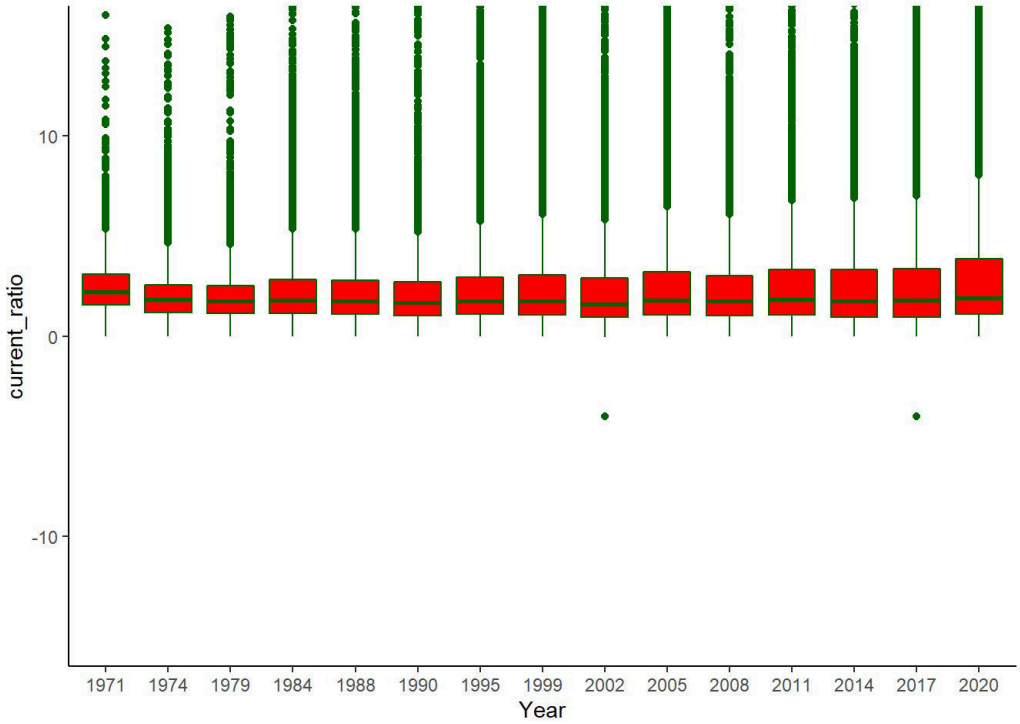
##Financial Ratios Table (Additional Table)

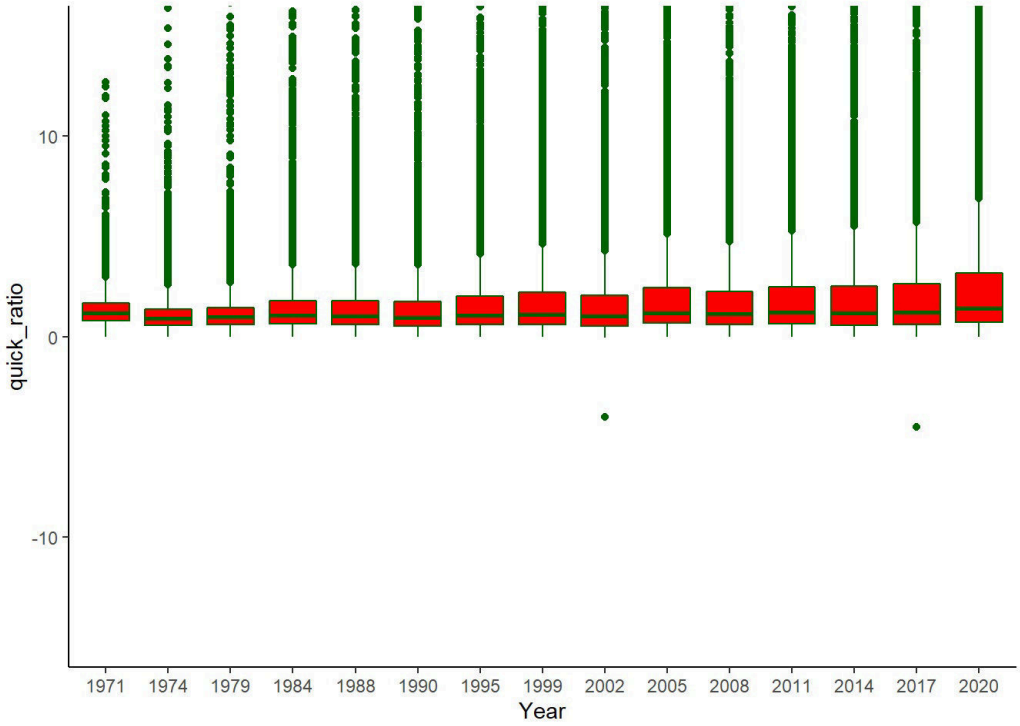
```
"Financial Ratios"
```

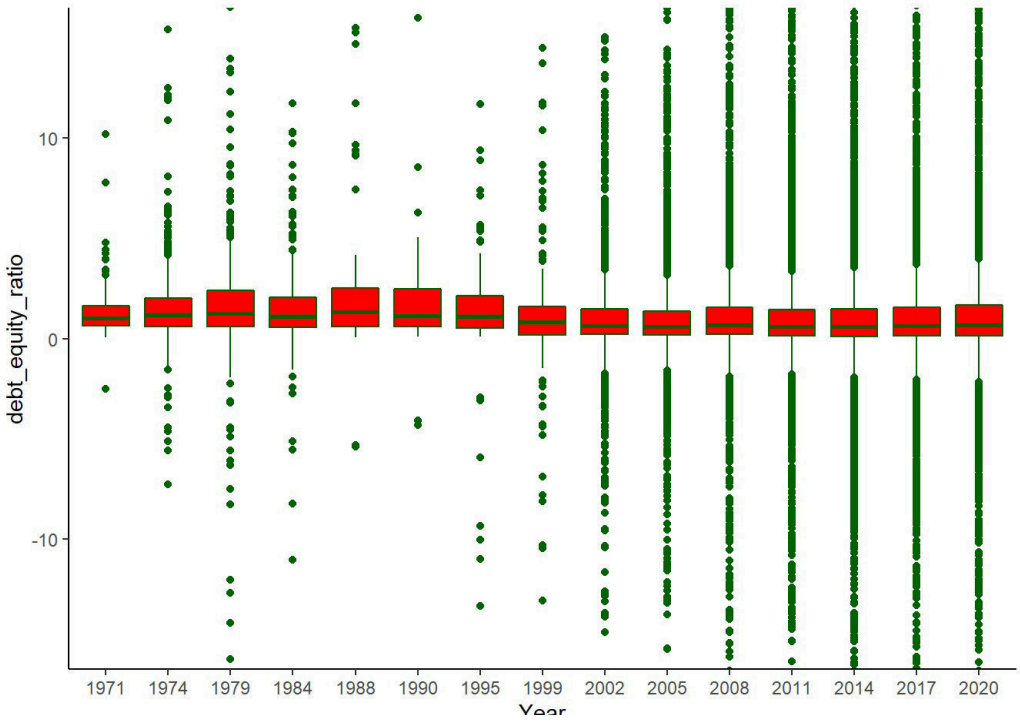
```
## [1] "Financial Ratios"
```

```
#Here we take all the data over all the years not just the selected list of years as taken previously as when we
take lag then it should calculate over previous year which will not happen if we discontinuity in the data


#Performing the similar operations of dropping AT where values are 0 and sorting the data by gvkey and fyear and
later subsetting the data for only selected years
all_data = subset_data%>%
  merge(gdp_deflator[c("fyear","multiplier")],by = "fyear")

all_data[,7:(ncol(all_data)-1)] = all_data[,7:(ncol(all_data)-1)]*unlist(all_data["multiplier"])

all_data = all_data[all_data["at"]!=0,]
all_data = all_data%>%
  select(-multiplier)

all_data[is.na(all_data)] = 0

Financial_Ratios_Data = all_data %>%
  group_by(gvkey,fyear)%>%
  mutate(current_ratio = if_else(lct!=0, act/lct,NA_real_),
         quick_ratio = if_else(lct!=0,(che+rect)/lct,NA_real_),
         debt_equity_ratio = if_else(teq!=0,lt/teq,NA_real_),
         dso = if_else(sale!=0,((rect + if_else(is.na(lag(rect)),0,lag(rect)))*0.5/sale)*365,NA_real_),
         dio = if_else(cogs!=0,((invt + if_else(is.na(lag(invt)),0,lag(invt)))*0.5/cogs)*365,NA_real_),
         dpo = if_else(cogs!=0,((ap + if_else(is.na(lag(ap)),0,lag(ap)))*0.5/cogs)*365,NA_real_),
         cash_conversion_cycle = dio + dso - dpo,
         total_asset_turnover = if_else((at + if_else(is.na(lag(at)),0,lag(at)))!=0,sale/((at + if_else(is.na(lag(
at)),0,lag(at)))*0.5),NA_real_),
         inventory_turnover = if_else((invt + if_else(is.na(lag(invt)),0,lag(invt)))!=0,cogs/((invt + if_else(is.n
a(lag(invt)),0,lag(invt)))*0.5),NA_real_),
         receivable_turnover = if_else((rect + if_else(is.na(lag(rect)),0,lag(rect)))!=0,sale/((rect + if_else(is.
na(lag(rect)),0,lag(rect)))*0.5),NA_real_),
         interest_burden = if_else(oiadp!=0,(oiadp-xint)/oiadp,NA_real_),
         interest_coverage = if_else(xint!=0,oiadp/xint,NA_real_),
         leverage = if_else(teq!=0,at/teq,NA_real_),
         roa = if_else((at + if_else(is.na(lag(at)),0,lag(at)))!=0,oiadp/((at + if_else(is.na(lag(at)),0,lag(at)))
*0.5),NA_real_),
         roe = if_else((teq + if_else(is.na(lag(teq)),0,lag(teq)))!=0,ni/((at + if_else(is.na(lag(teq)),0,lag(teq)
))*0.5),NA_real_),
         ros = if_else(sale!=0,oiadp/sale,NA_real_))%>%
  ungroup()%>%
  filter(fyear %in% year_list)

liquidity_solvency_ratios = c("current_ratio","quick_ratio","debt_equity_ratio")
activity = c("dso","dio","dpo","cash_conversion_cycle")
asset_utilization = c("total_asset_turnover","inventory_turnover","receivable_turnover")
leverage_ratio = c("interest_burden","interest_coverage","leverage")
profitability_ratio = c("roa","roe","ros")

#calculating the different ratios for each of the tabs in the excel file(  additional table)
Liquidity_Solvency = Financial_Ratios_Data%>%
  select(c(fyear,liquidity_solvency_ratios))

Activity = Financial_Ratios_Data%>%
  select(c(fyear,activity))

Asset_Utilization = Financial_Ratios_Data%>%
  select(c(fyear,asset_utilization))

Leverage = Financial_Ratios_Data%>%
  select(c(fyear,leverage_ratio))
```

```
Profitability = Financial_Ratios_Data%>%
  select(c(fyear,profitability_ratio))

#Calcualting the descriptive stats for all of the above datasets
c(count_liquidity_solvency_data,mean_liquidity_solvency_data,p25_liquidity_solvency_data,p50_liquidity_solvency_d
ata, p75_liquidity_solvency_data, std_liquidity_solvency_data,max_liquidity_solvency_data,min_liquidity_solvency_
data)%<-%stats(Liquidity_Solvency)

c(count_activity,mean_activity,p25_activity,p50_activity, p75_activity, std_activity,max_activity,min_activity)%<
-%stats(Activity)

c(count_asset_utiltization,mean_asset_utiltization,p25_asset_utiltization,p50_asset_utiltization, p75_asset_utilt
ization, std_asset_utiltization,max_asset_utiltization,min_asset_utiltization)%<-%stats(Asset_Utilization)

c(count_leverage,mean_leverage,p25_leverage,p50_leverage, p75_leverage, std_leverage,max_leverage,min_leverage)%<
-%stats(Leverage)

c(count_profitability,mean_profitability,p25_profitability,p50_profitability, p75_profitability, std_profitabilit
y,max_profitability,min_profitability)%<-%stats(Profitability)

#Plotting the box plots for the same
plot_all_boxplots(Liquidity_Solvency)
```

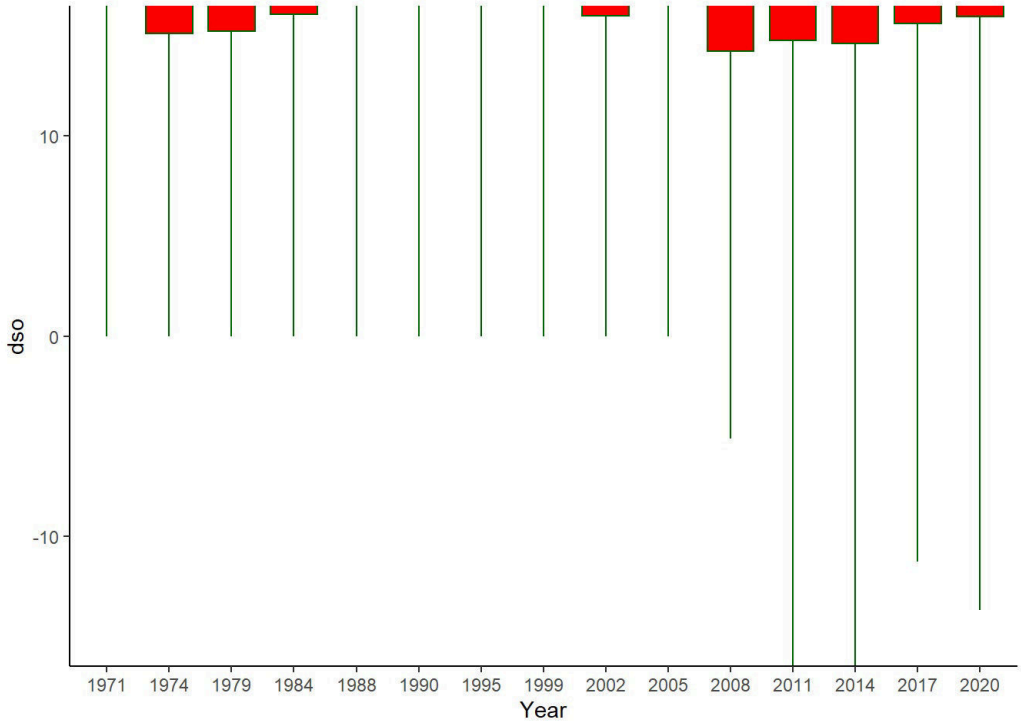

current_ratio
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

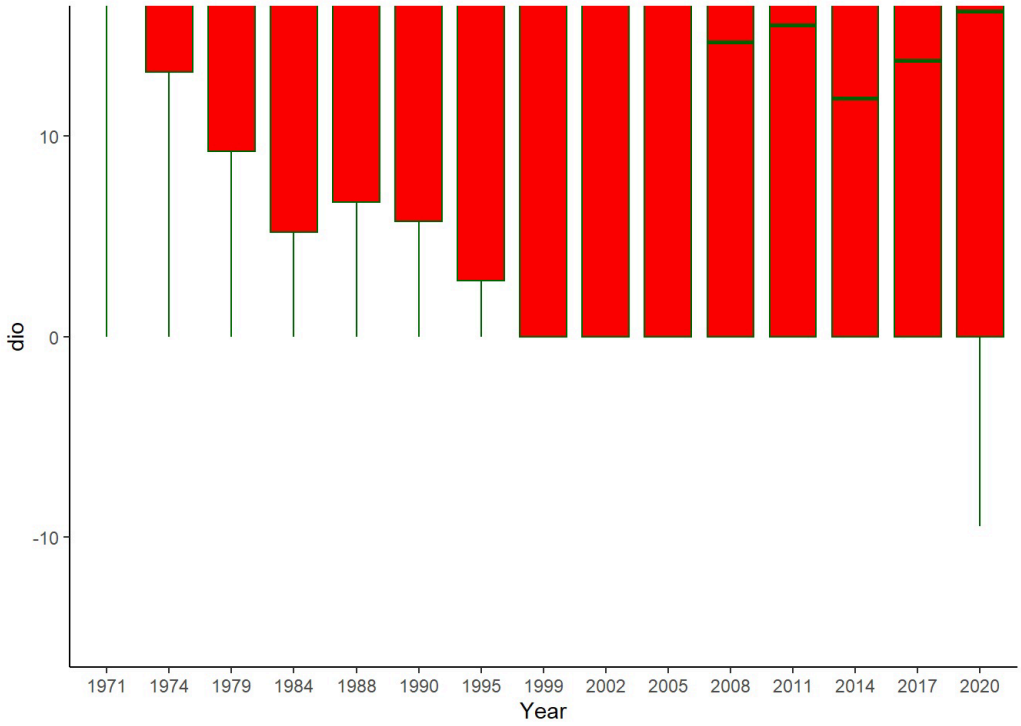

quick_ratio
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

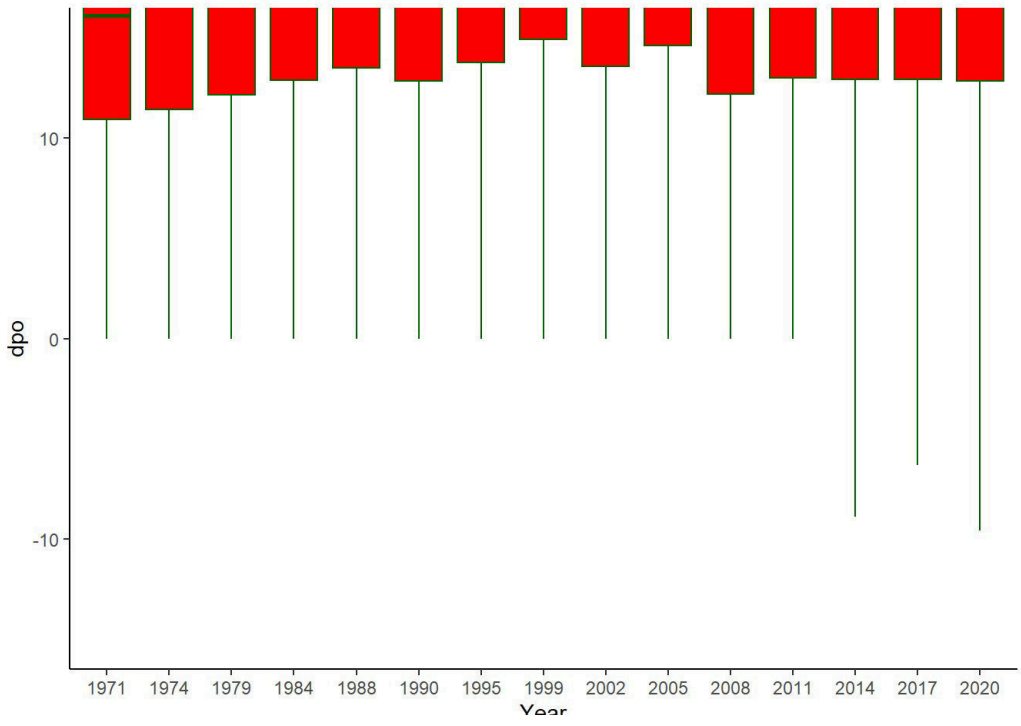

debt_equity_ratio
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

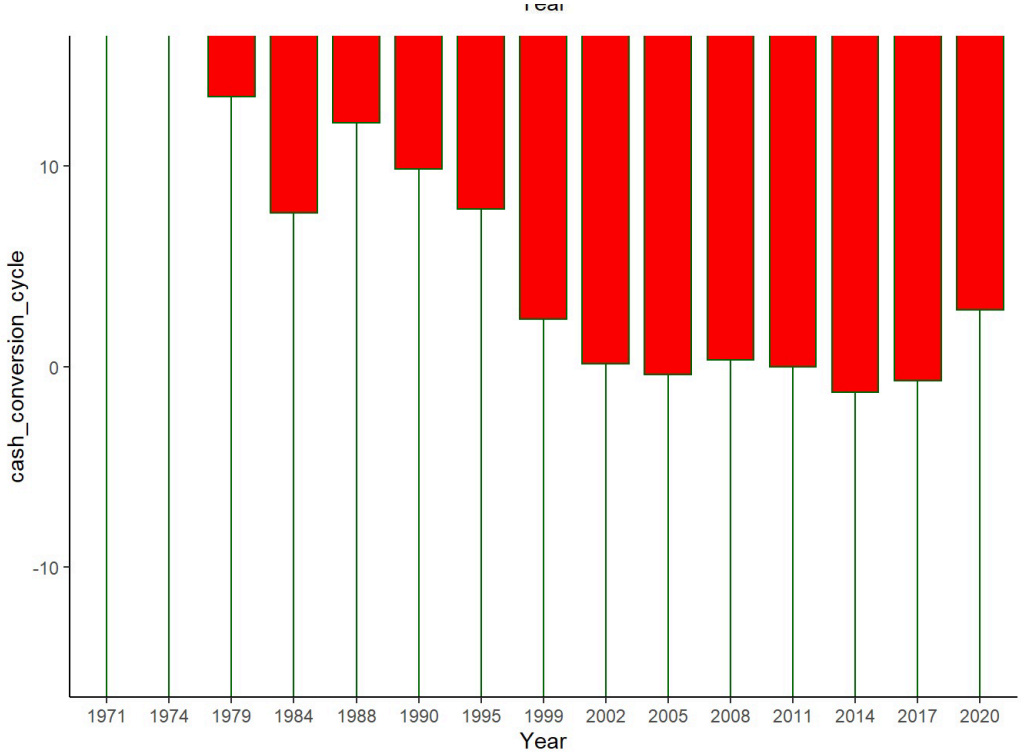
Year

```
plot_all_boxplots(Activity)
```

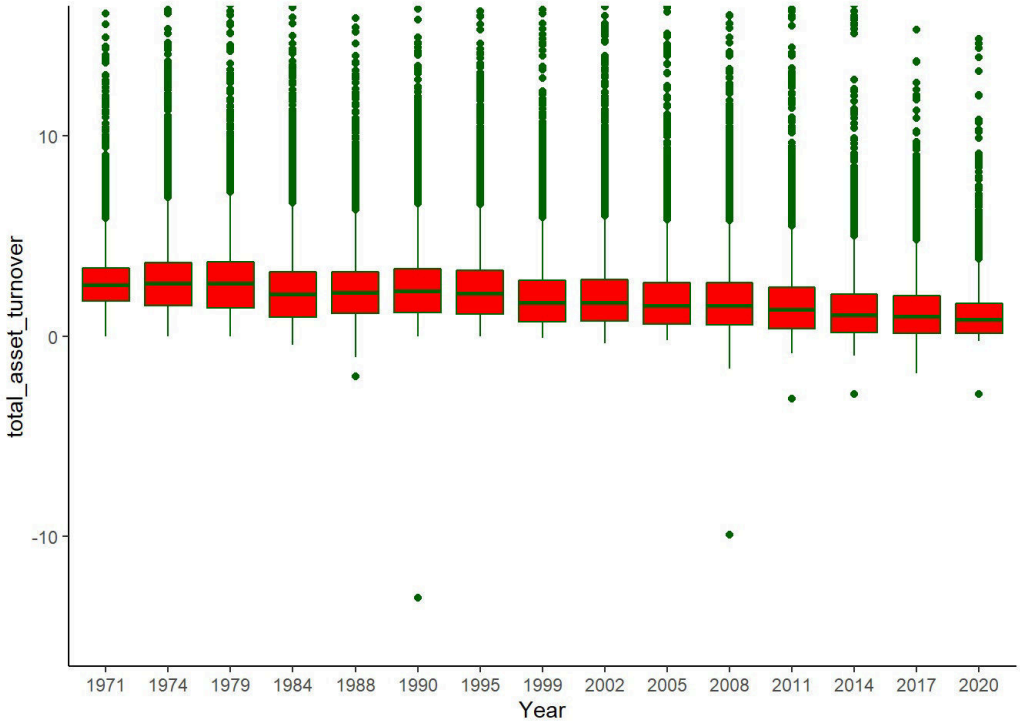

dso
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

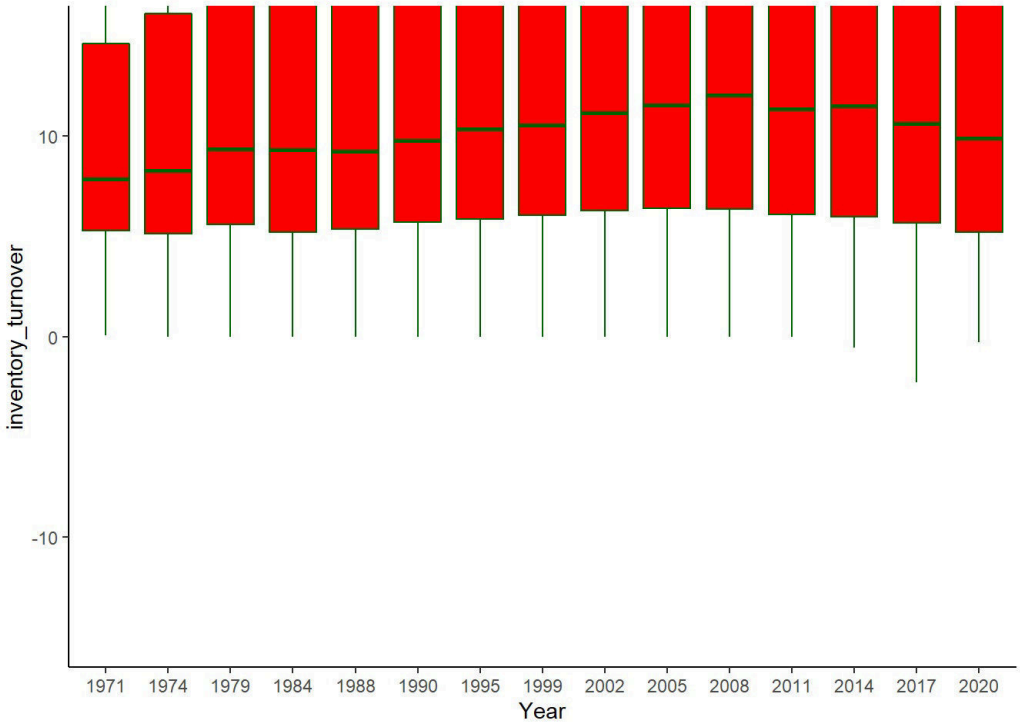

dio
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

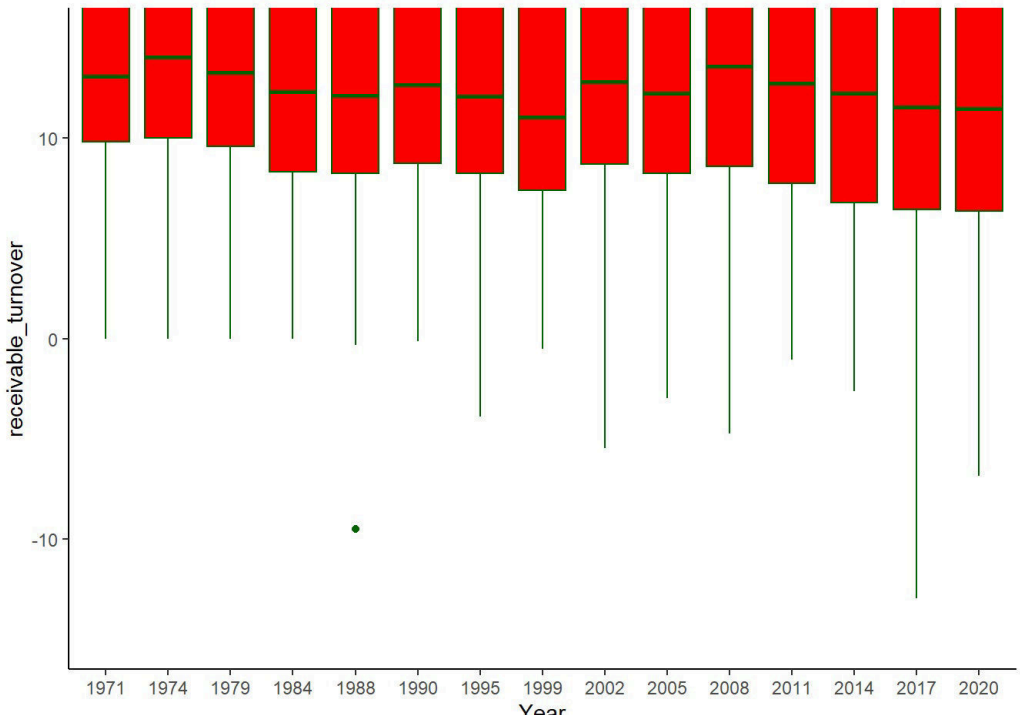

dpo
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year




```
plot_all_boxplots(Asset_Utilization)
```

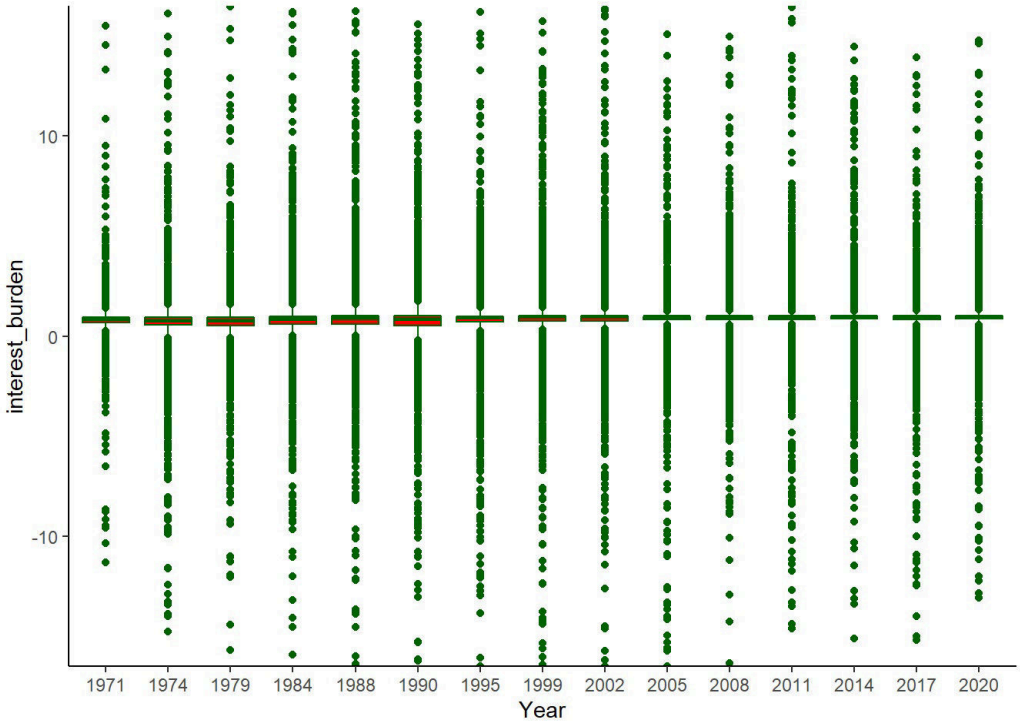

total_asset_turnover
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

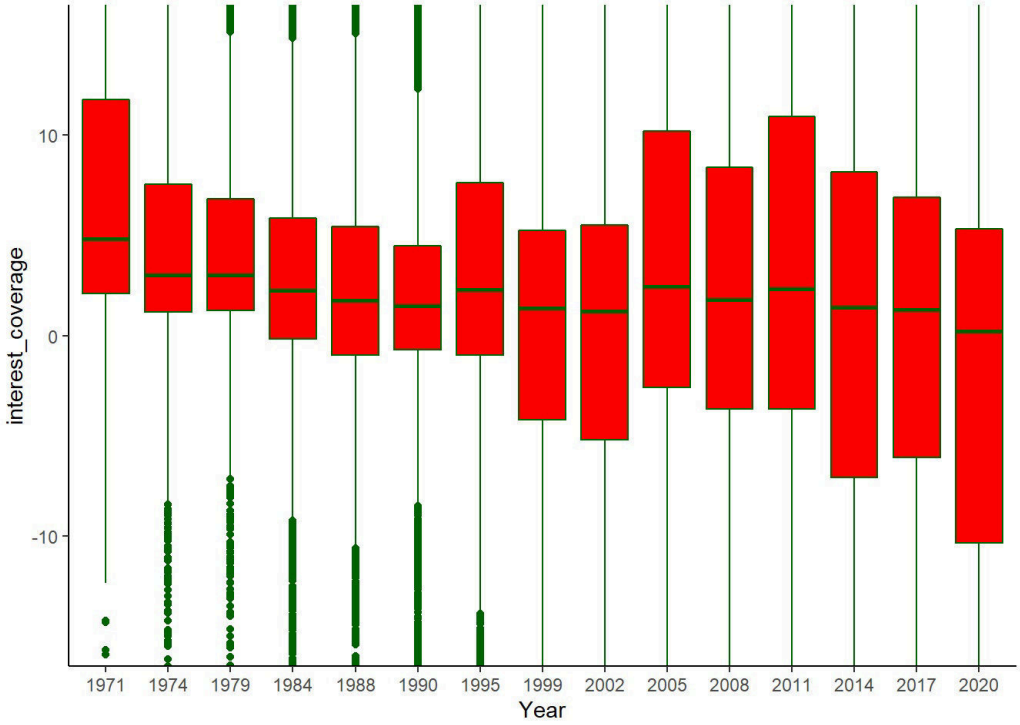

inventory_turnover
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

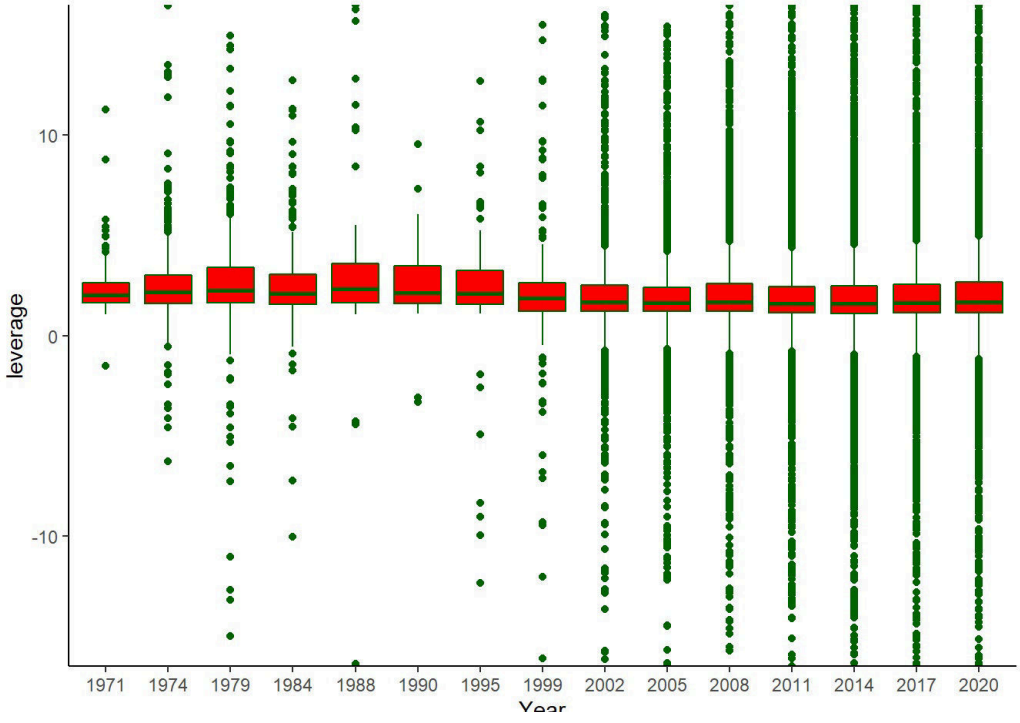

receivable_turnover
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

Year

```
plot_all_boxplots(Leverage)
```

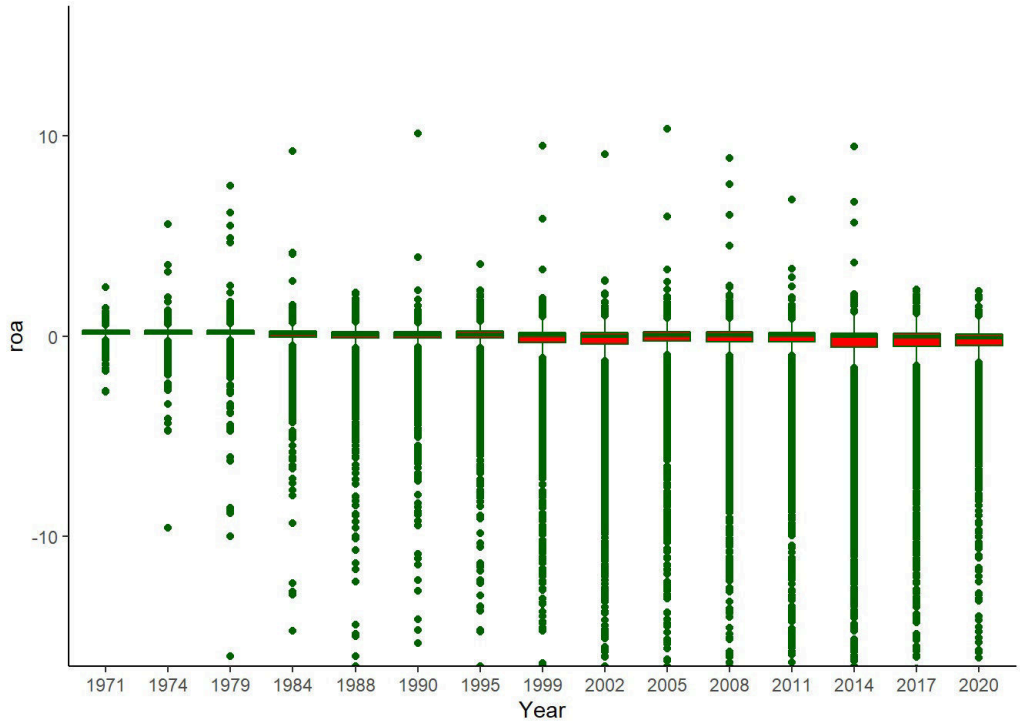

interest_burden
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

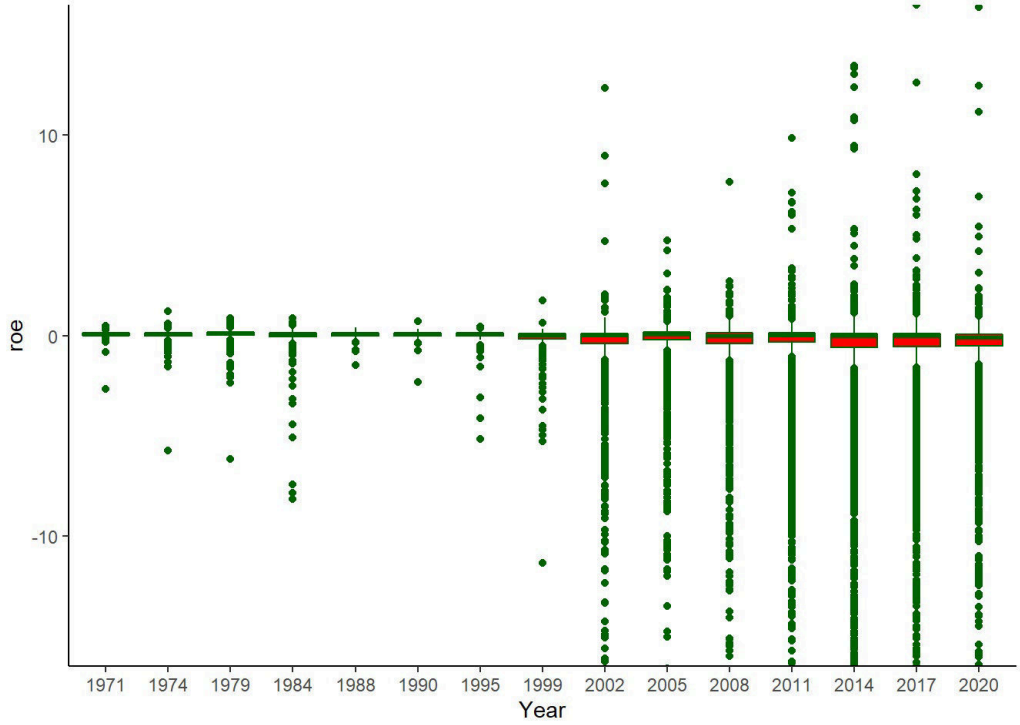

interest_coverage
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

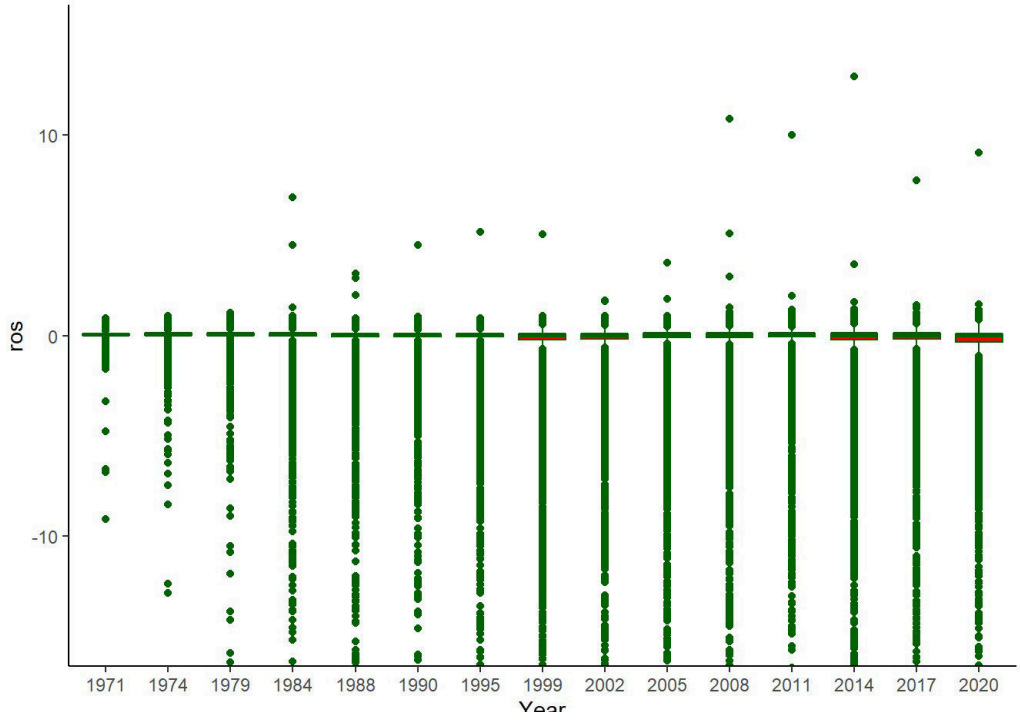

leverage
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

Year

```
plot_all_boxplots(Profitability)
```


roa
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


roe
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


ros
10
0
-10
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

Year

##NBER Recession Data

```
"Loading the NBER Recession Data from FRED and merging with our dataframe"
```

```
## [1] "Loading the NBER Recession Data from FRED and merging with our dataframe"
```

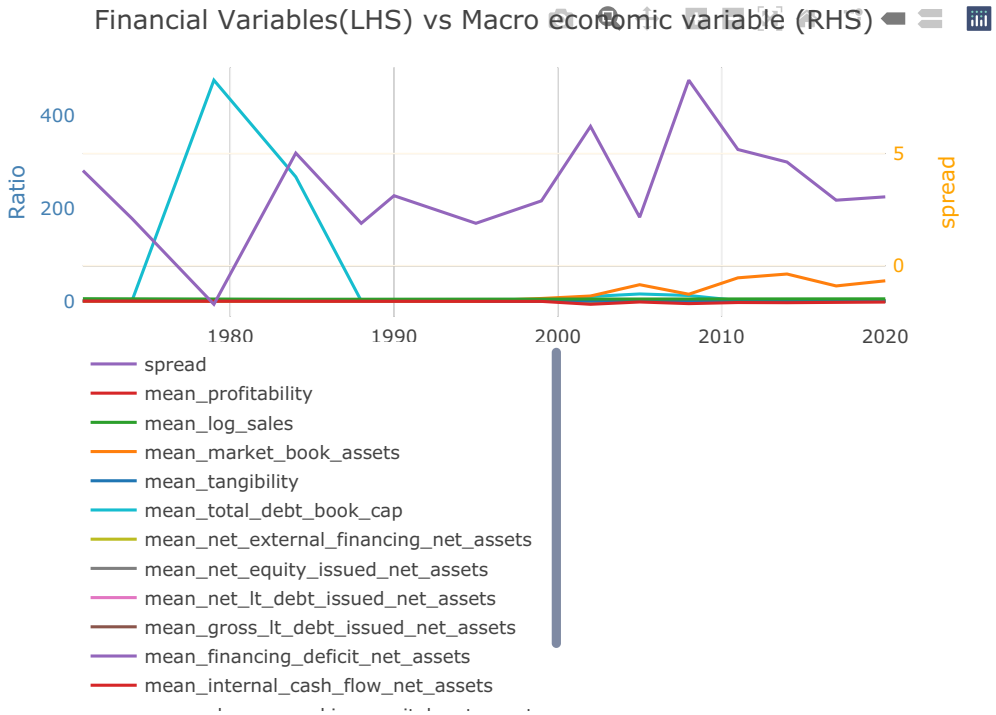
```
recession_years = "USREC" %>%
  tq_get(get = "economic.data", from = "1950-01-01") %>%
  select(date, recession_flag = price) %>%
  mutate(fyear = case_when(recession_flag == 1 & lag(recession_flag == 0) ~ str_glue("{year(date)}"),
         TRUE ~ NA_character_),fyear = as.numeric(fyear)) %>%
  filter(recession_flag == 1) %>%
  fill(fyear, .direction = "down") %>%
  group_by(fyear) %>%
  mutate(start = case_when(date == min(date) ~ date,
                           TRUE ~ NA_Date_),
         end = case_when(date == max(date) ~ date,
                         TRUE ~ NA_Date_),
         end = lead(end)
         ) %>%
  filter(!is.na(start)) %>%
  select(-c(date,start,end))
#Attaching the data with our dataframe for recession and non-recession years
all_years = seq(1950, 2020, by=1)
all_years = data.frame(fyear = all_years)
recession_years = merge(all_years,recession_years,by = "fyear",all.x = TRUE)
recession_years[is.na(recession_years)] = 0
#Adding the years 1974 and 1982 to the recession years as in these years recession continued from the previous ye
ars
recession_years[(recession_years$fyear == 1974) | (recession_years$fyear == 1982),"recession_flag"] = 1


Recession_Df = merge(Financial_Ratios_Data,recession_years[recession_years$recession_flag == 1,],by = "fyear")
Non_Recession_Df = merge(Financial_Ratios_Data,recession_years[recession_years$recession_flag == 0,],by = "fyear"
)

#Calculating the descriptive stats for the Recession and Non-Recession years
c(count_rescession,mean_recession,p25_recession,p50_recession,p75_recession,std_recession,max_recession,min_reces
sion) %<-% stats(Recession_Df[c(1,87:(ncol(Recession_Df)-1))])

c(count_not_rescession,mean_not_recession,p25_not_recession,p50_not_recession,p75_not_recession,std_not_recession
,max_not_recession,min_not_recession) %<-% stats(Non_Recession_Df[c(1,87:(ncol(Non_Recession_Df)-1))])
```

##Macroeconomic Data Analysis (BAA Spread, Kansas Stress Index)

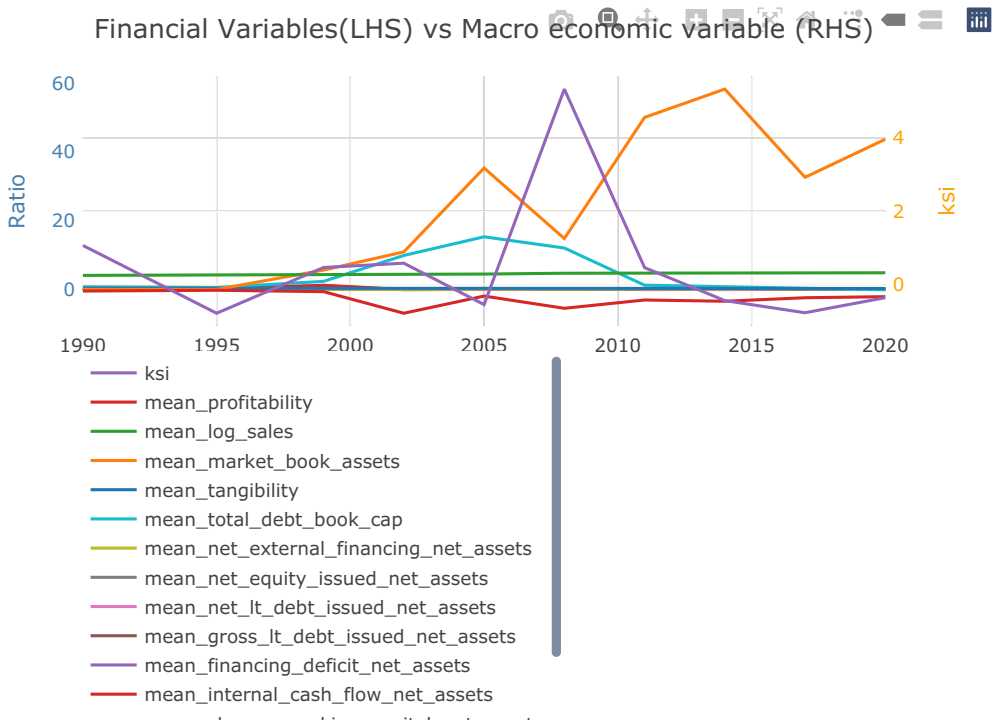
```
"Extracting data from FRED for the BAA spread and Kansas Stress Index for the time period and merging with the ti
meframe - Plotting the line charts with dual y-axis"
```

```
## [1] "Extracting data from FRED for the BAA spread and Kansas Stress Index for the time period and merging with
the timeframe - Plotting the line charts with dual y-axis"
```

```
#Get macroeconomic data which is used for all the parts
baa_spread <-
   "BAAFFM" %>%
  tq_get(get = "economic.data", from = "1950-01-01") %>%
  select(fyear = date, spread  =  price) %>%
  mutate(fyear = as.numeric(year(fyear)))%>%
  group_by(fyear)%>%
  slice(n())

kansas_stress_index<-
  "KCFSI" %>%
  tq_get(get = "economic.data", from = "1950-01-01") %>%
  select(fyear = date, ksi = price) %>%
  mutate(fyear = as.numeric(year(fyear)))%>%
  group_by(fyear)%>%
  slice(n())

BAA_Df = merge(mean_key_variables_t9,baa_spread,by = "fyear")

Stress_Df =  merge(mean_key_variables_t9,kansas_stress_index,by = "fyear")

plot_multiple_line_chart(BAA_Df,"Financial Variables","Ratio",1)
```




```
plot_multiple_line_chart(Stress_Df,"Financial Variables","Ratio",1)
```

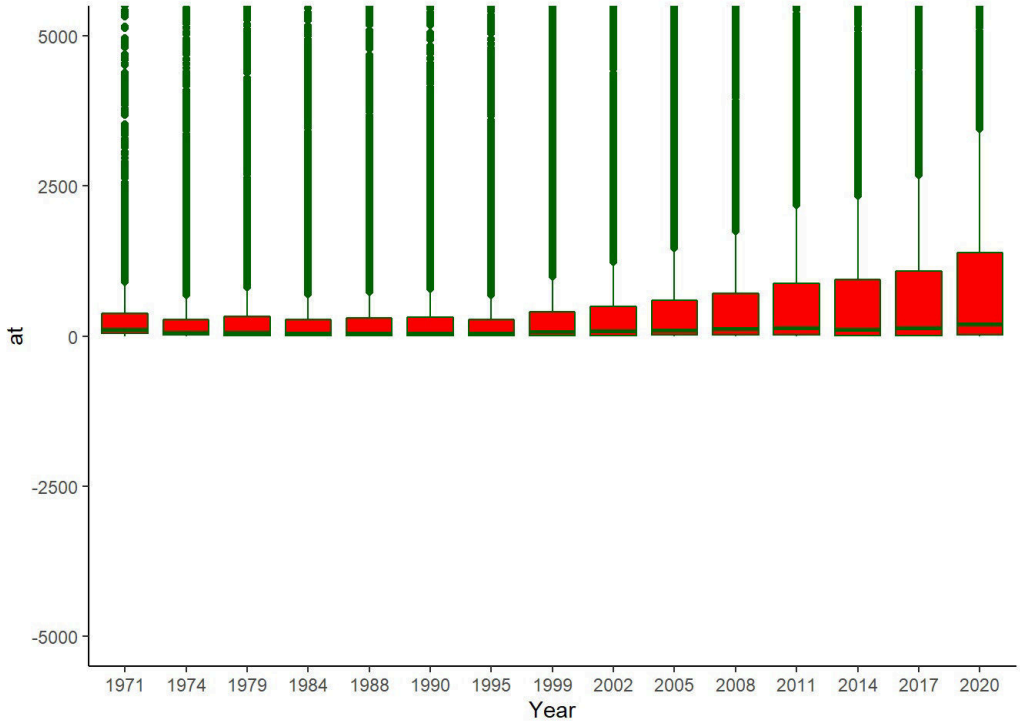


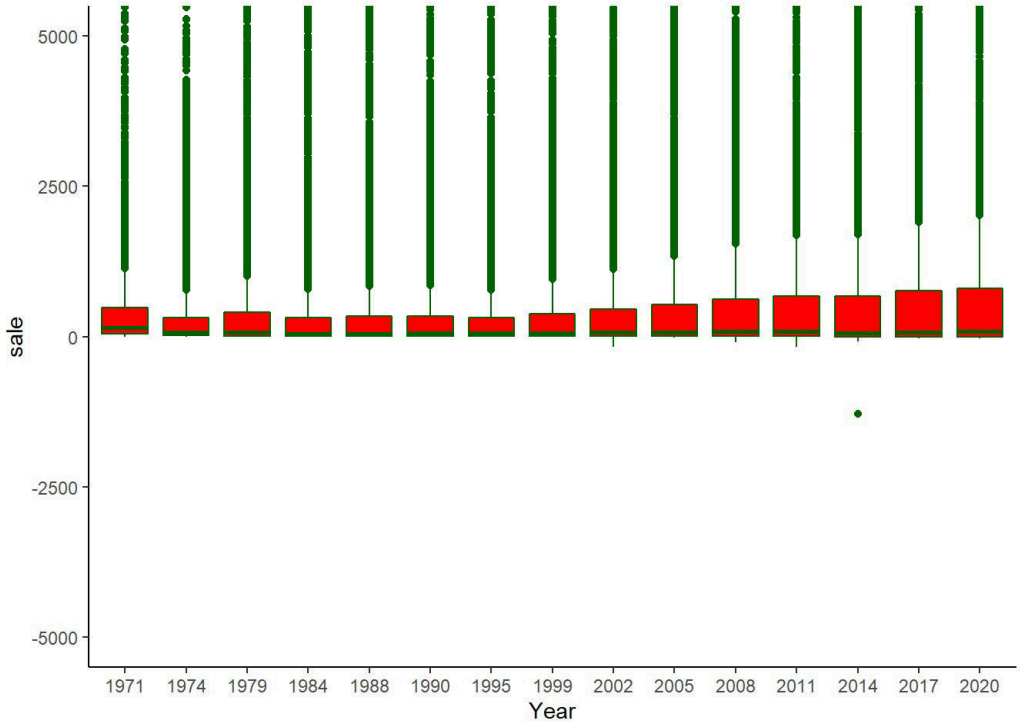
##1992 Dollars Terms Analysis for Table 9

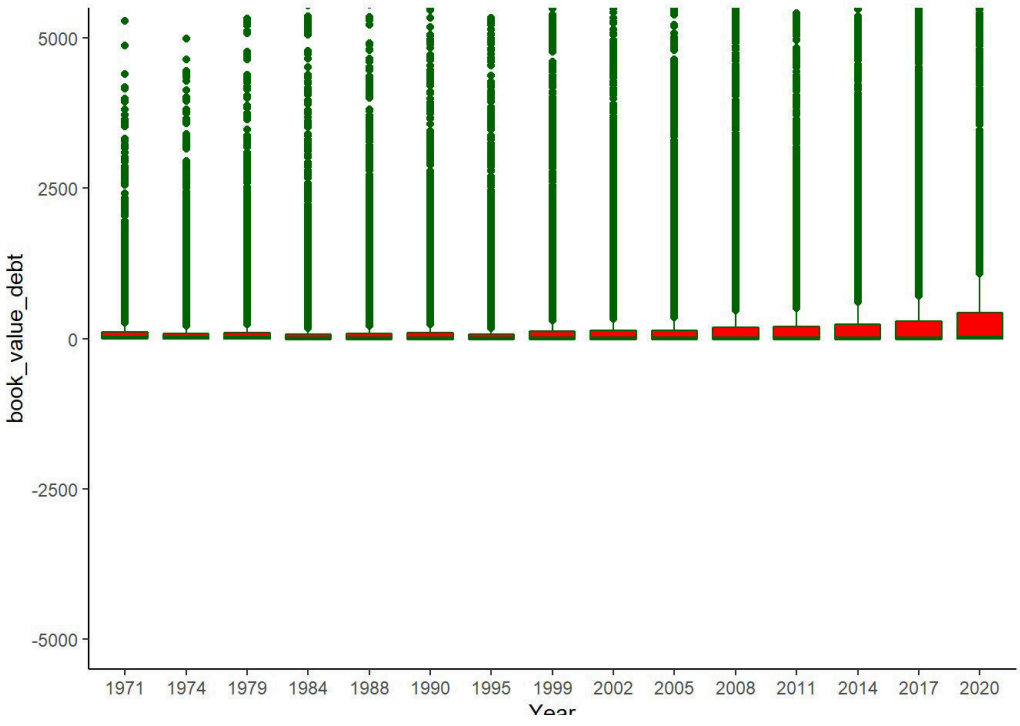
```
"Creating and using old varibales in previous tables to create new variables for Table 9 descriptive stats and Bo
x plots"
```

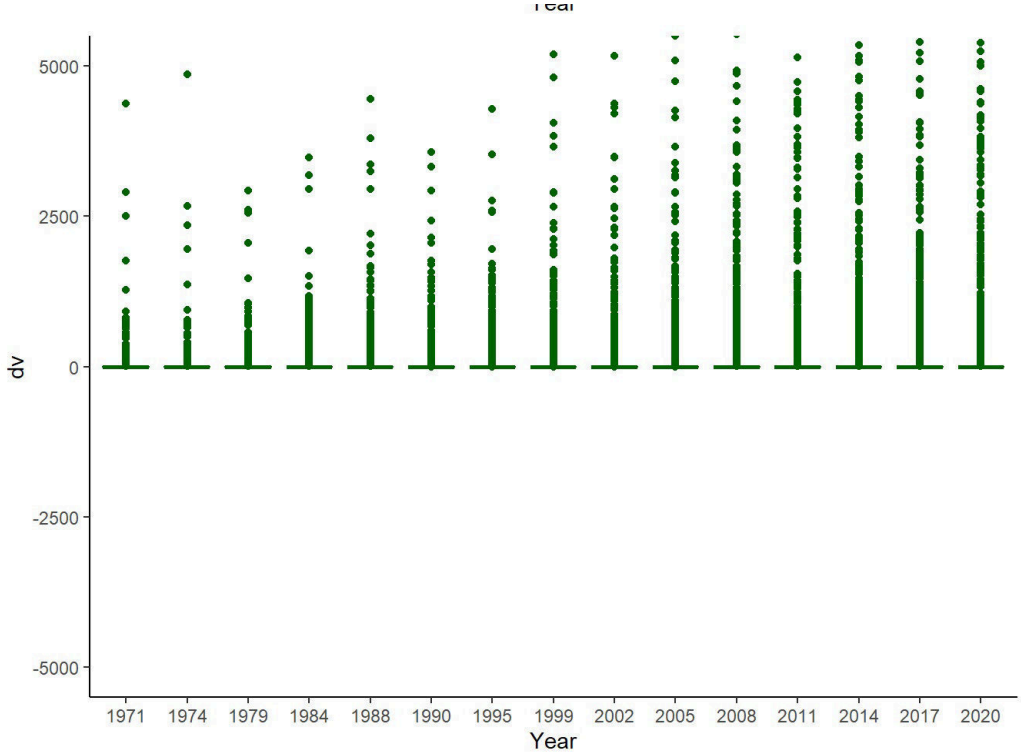
```
## [1] "Creating and using old varibales in previous tables to create new variables for Table 9 descriptive stats
and Box plots"
```

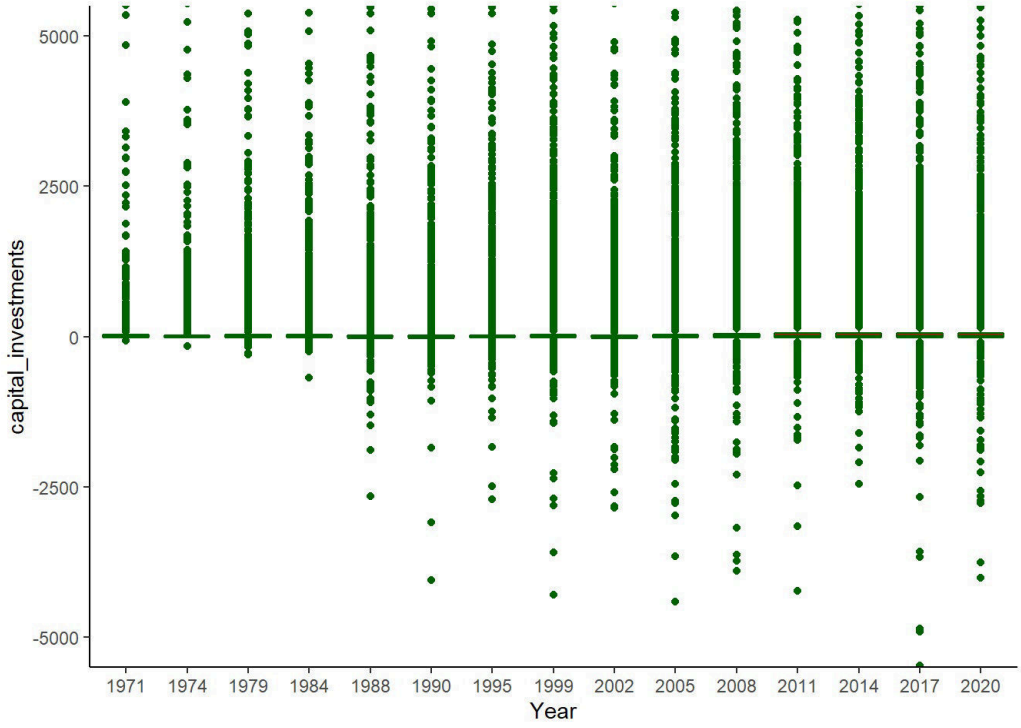
```
Dollar_Terms_t9 = Modified_Data %>%
  mutate(at = at,
         sale = sale,
         book_value_debt = dlc+dltt,
         dv = dv,
         capital_investments = investment_net_assets*(at-lct),
         change_working_capital = change_working_capital_net_assets*(at-lct),
         internal_cash_flow = internal_cash_flow_net_assets*(at-lct),
         financing_deficit = financing_deficit_net_assets*(at-lct),
         long_term_debt_issuance = dltis,
         reduction_long_term_debt = dltr,
         net_debt_issued = long_term_debt_issuance - reduction_long_term_debt,
         common_stock_sale = sstk,
         common_stock_purchase = prstkc,
         net_equity_issued = common_stock_sale - common_stock_purchase,
         net_external_financing = net_debt_issued + net_equity_issued)

#Replace NA values back for initial variables for Balance sheet, income statement and cash flow statement because
we want to keep those NA values which we have replace earlier by 0 in previous operation. This is done to calcual
te precisely the values of descriptive stats
Dollar_Terms_t9_NA = copy(Dollar_Terms_t9)
for (i in (1:ncol(na_values))){
  if(any(na_values[,i])){
    Dollar_Terms_t9_NA[,i] = replace(as.vector(unlist(Dollar_Terms_t9_NA[,i])), na_values[,i] == TRUE, NA)
  }
}

Dollar_Terms_t9_NA = Dollar_Terms_t9_NA %>%

select(fyear,at,sale,book_value_debt,dv,capital_investments,change_working_capital,internal_cash_flow,financing_d
eficit,long_term_debt_issuance,reduction_long_term_debt,net_debt_issued,common_stock_sale,common_stock_purchase,n
et_equity_issued,net_external_financing)

#Calculating the descriptive stats
c(count_dollar_T9,mean_dollar_T9,p25_dollar_T9,p50_dollar_T9, p75_dollar_T9, std_dollar_T9,max_dollar_T9,min_doll
ar_T9)%<-%stats(Dollar_Terms_t9_NA)

#Plot the box plots
plot_all_boxplots(Dollar_Terms_t9_NA,1)
```

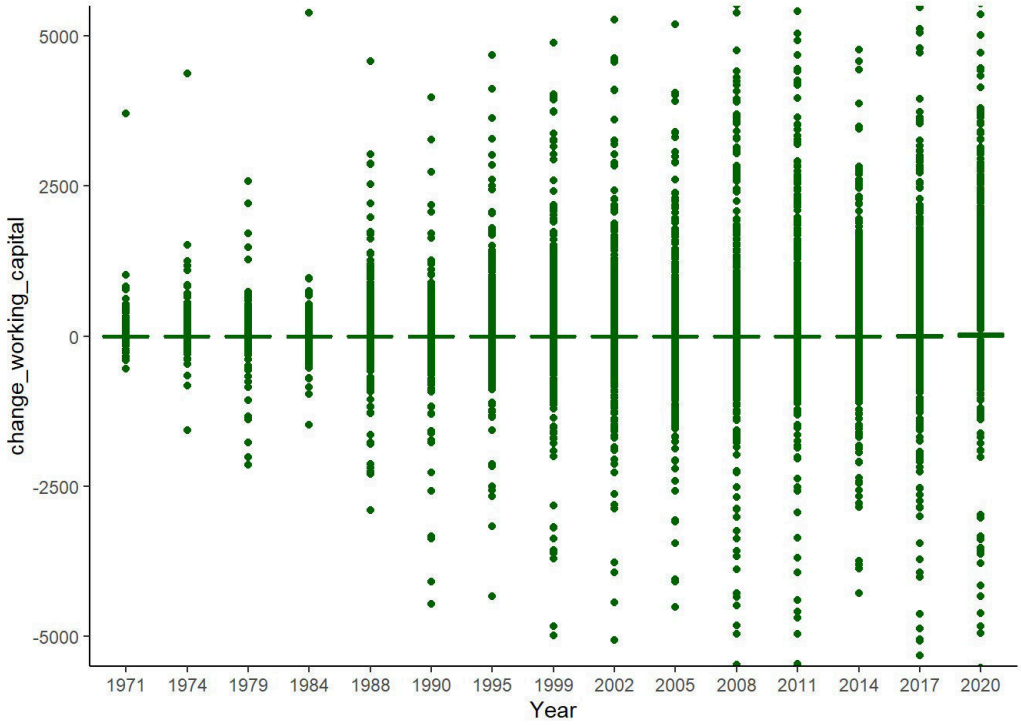

at
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

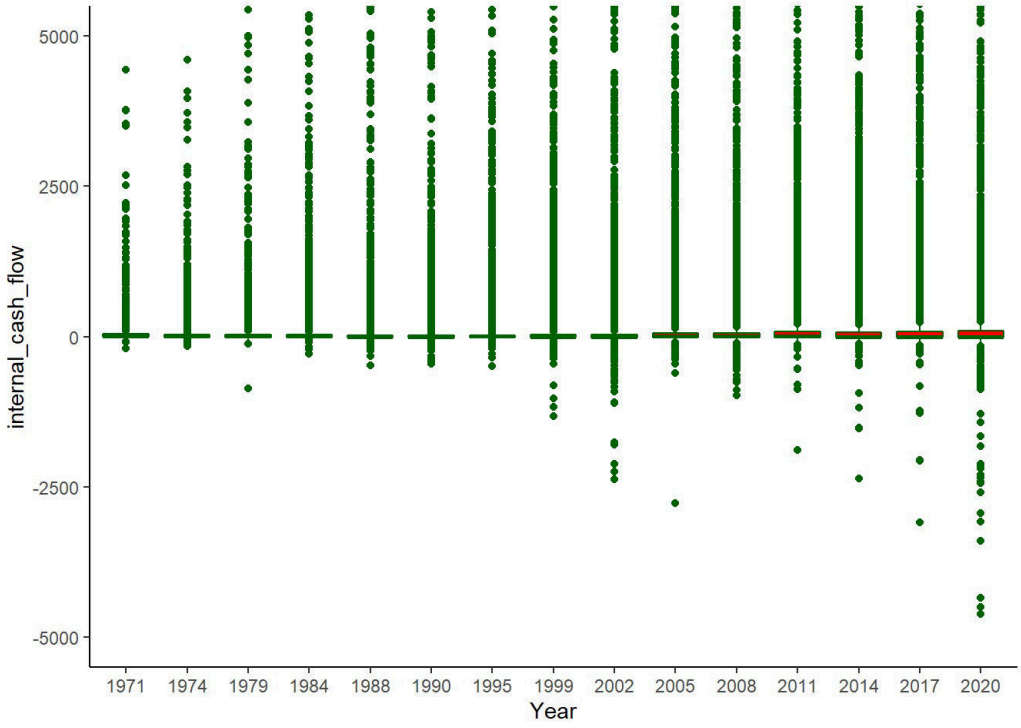

sale
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

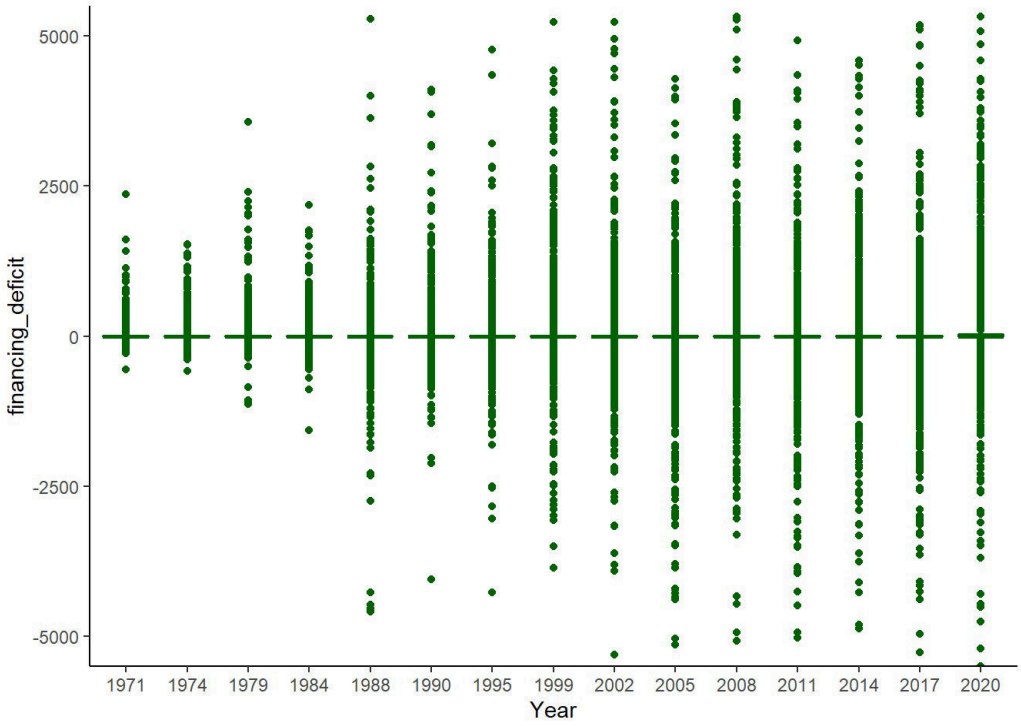

book_value_debt
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

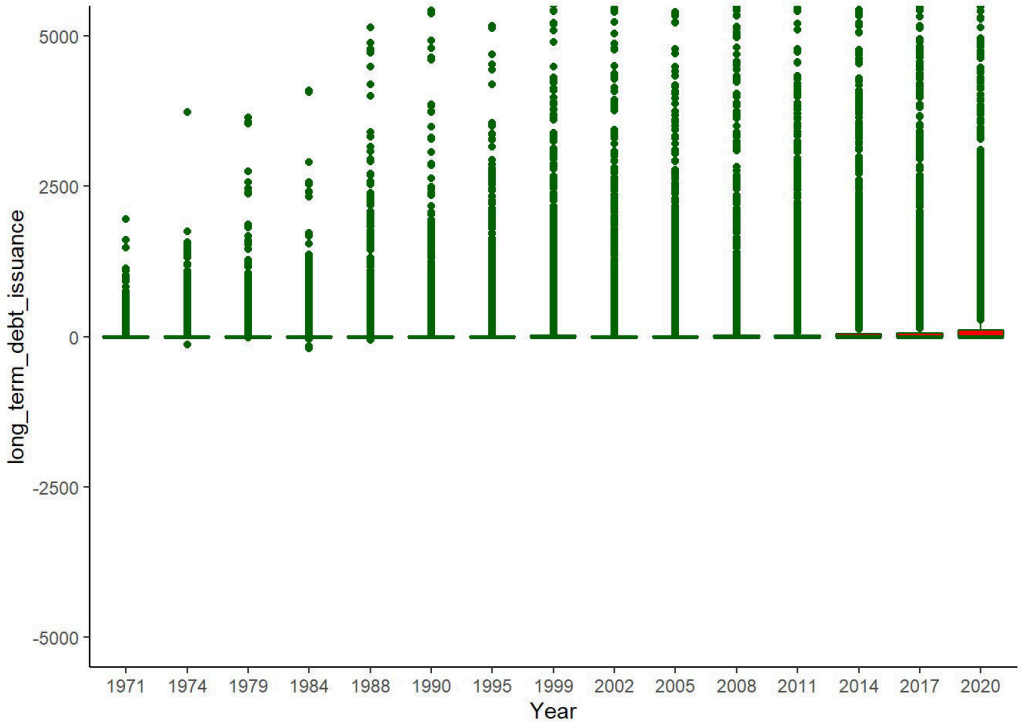

dv
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

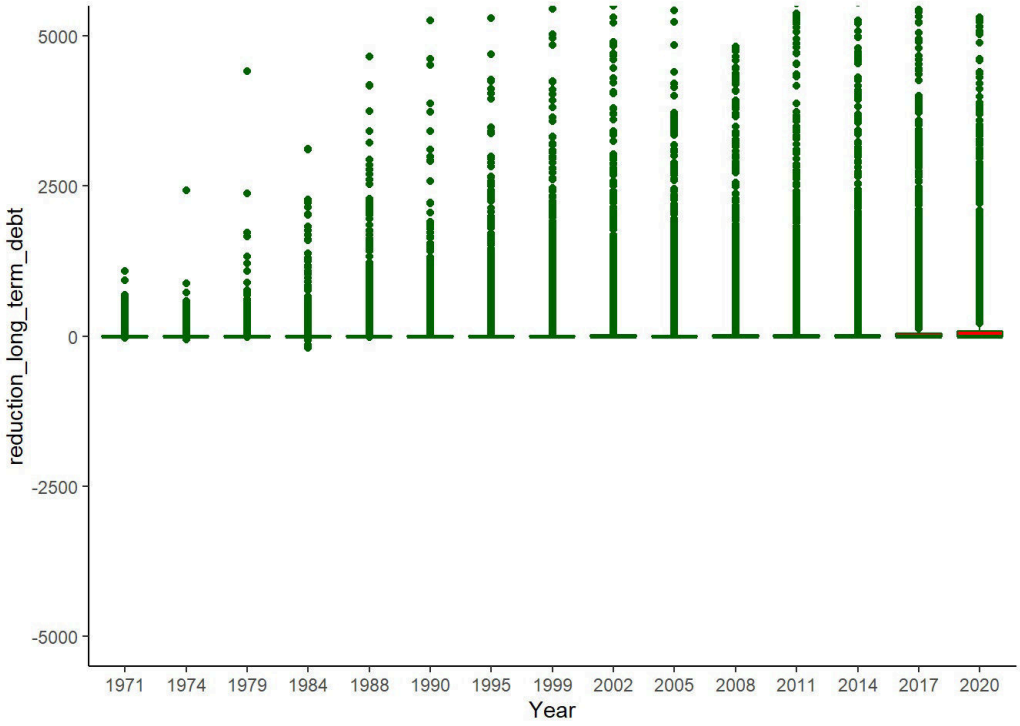

capital_investments
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

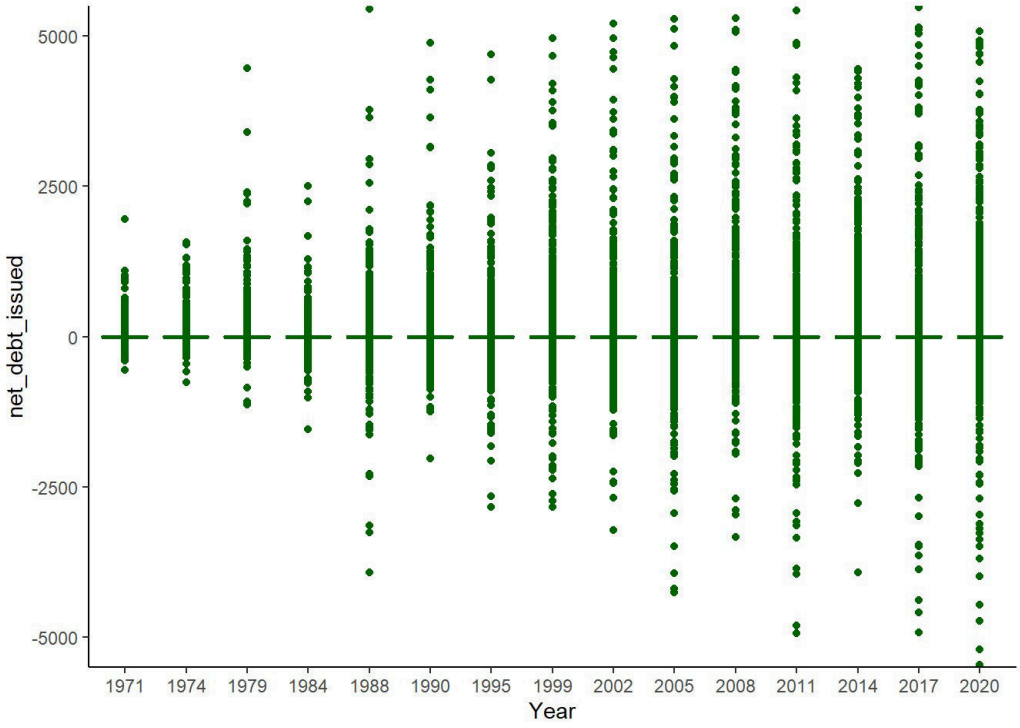

change_working_capital
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

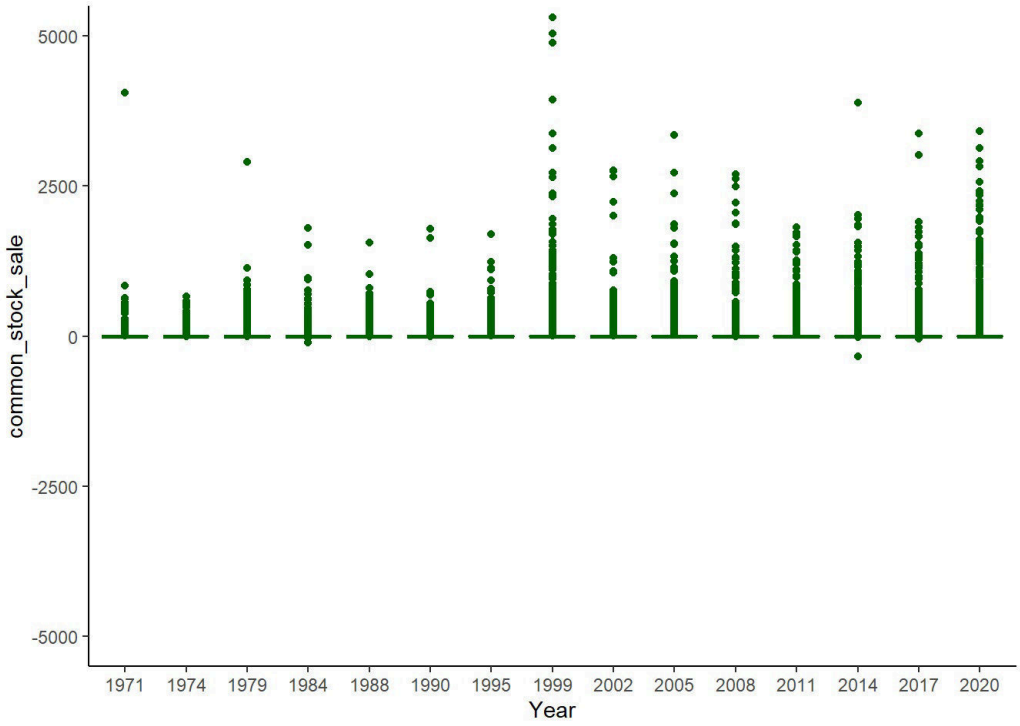

internal_cash_flow
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

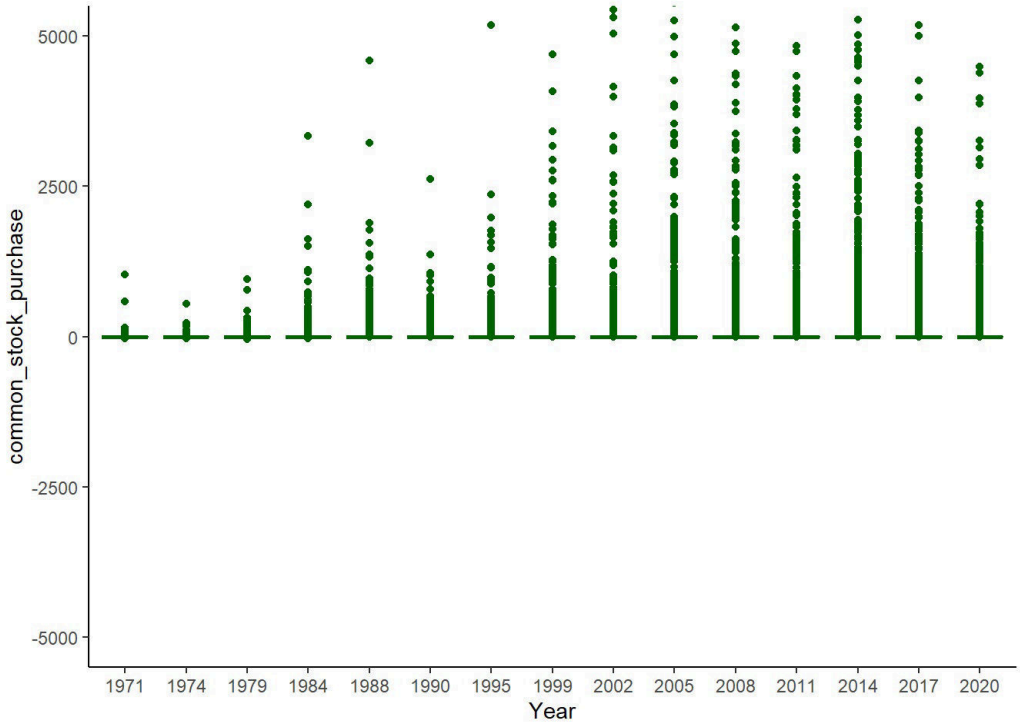

financing_deficit
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

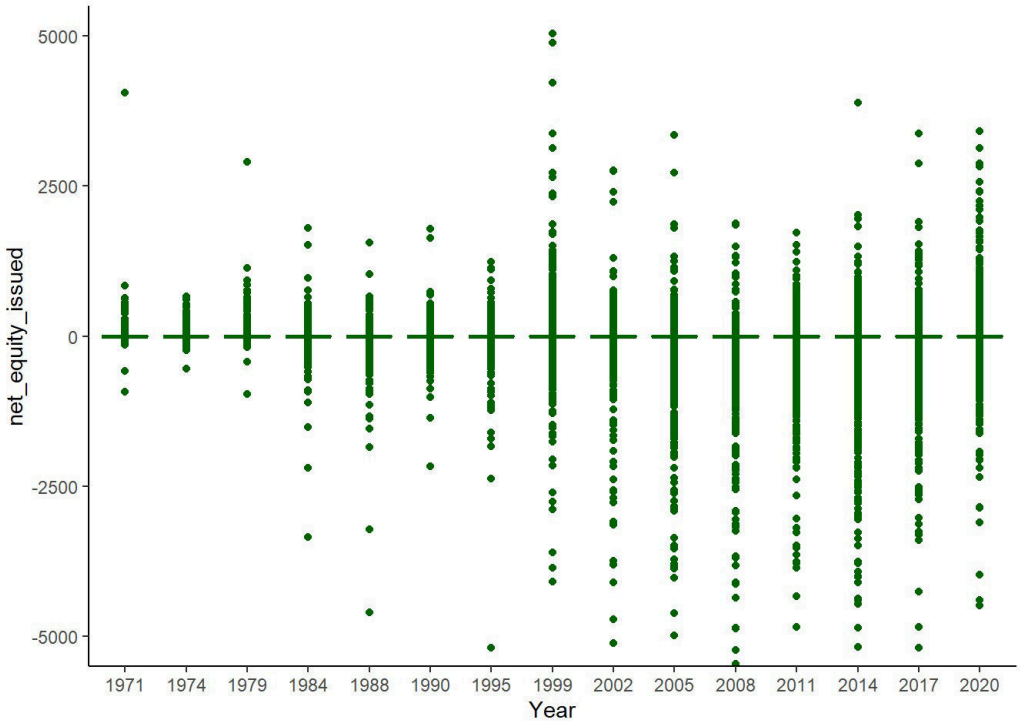

long_term_debt_issuance
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

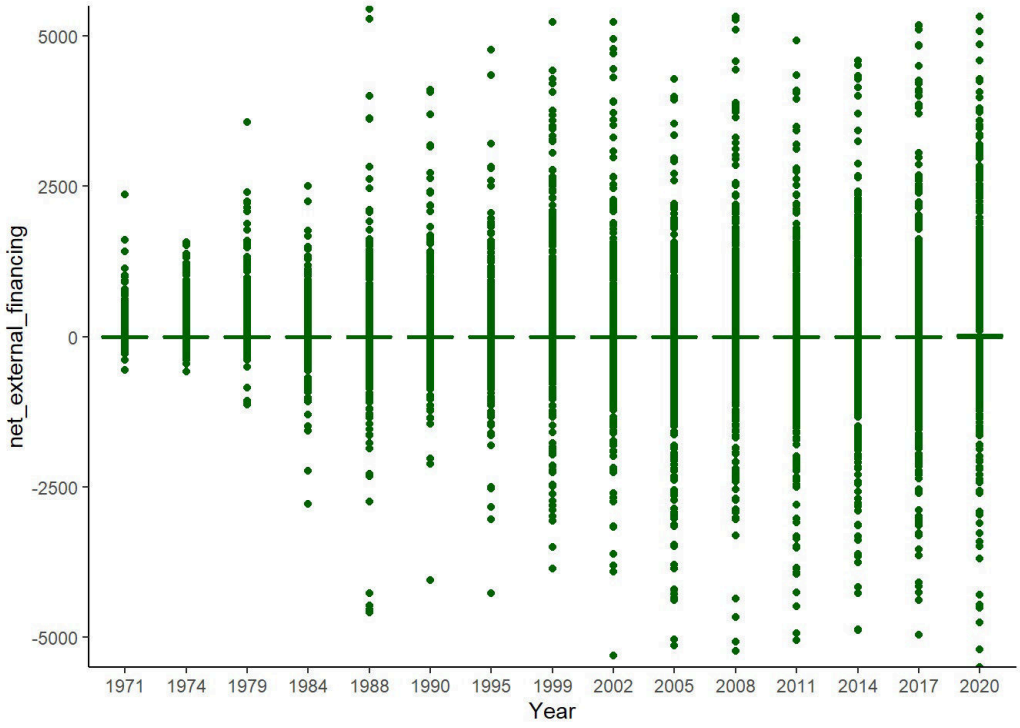

reduction_long_term_debt
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


net_debt_issued
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


common_stock_sale
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


common_stock_purchase
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


net_equity_issued
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year


net_external_financing
5000
2500
0
-2500
-5000
1971
1974
1979
1984
1988
1990
1995
1999
2002
2005
2008
2011
2014
2017
2020
Year

```
# dev.off()
```