```vba
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''Filtering - Data Exploitation for Prices Comparison'''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''Please refer to the R file associated for the results of the''''''''''
'''''''''''''''''''''''''''''''''''''Prices Comparison'''''''''''''''''''''''''''''''''''

Option Explicit

Sub Code_filtering_data()

ActiveWorkbook.Worksheets("Original_Data").Activate

'we declare the variables i and j which are going to be used in for loops
Dim i As Long
Dim j As Long
Dim k As Long

'We identify the names of the columns which are going to be used later
Dim Units_col As Integer
Dim Init_acq_cost_col As Integer
Dim PropertySubtype_col As Integer
Dim CapRate_col As Integer
Dim Year_col As Integer
Dim NetSalePrice_col As Integer
Dim Prop_col As Integer
Dim Year_built_col As Integer
Dim PropertyType_col As Long
Dim Quarter_col As Long

'we declare the ranges which are going to be used for the identification
Dim Rng1 As Range
Dim Rng2 As Range
Dim Rng3 As Range
Dim Rng4 As Range
Dim Rng5 As Range
Dim Rng6 As Range
Dim Rng7 As Range
Dim Rng8 As Range
Dim Rng9 As Range

'These strings will be looked for on the main table with the data
Dim units As String
Dim InitAcqCost As String
Dim PropertyType As String
Dim CapRate As String
Dim Year As String
Dim NetSalePrice As String
Dim Prop As String
Dim YrBuilt As String
Dim Quarter As String

'we select the first row
With ActiveSheet.Range("A1:BU1")

'we find the names in the range just declared - first row of the original data
    Set Rng1 = .Find(What:="units", _
                        After:=.Cells(.Cells.Count), _
                        LookIn:=xlValues, _
                        LookAt:=xlWhole, _
                        SearchOrder:=xlByRows, _
                        SearchDirection:=xlNext, _
                        MatchCase:=False)

    Set Rng2 = .Find(What:="InitAcqCost", _
                        After:=.Cells(.Cells.Count), _
                        LookIn:=xlValues, _
                        LookAt:=xlWhole, _
                        SearchOrder:=xlByRows, _
                        SearchDirection:=xlNext, _
```

```
                              MatchCase:=False)

      Set Rng3 = .Find(What:="PropertyType", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng4 = .Find(What:="CapRate", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng5 = .Find(What:="Year", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng6 = .Find(What:="NetSalePrice", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng7 = .Find(What:="PROP", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng8 = .Find(What:="YrBuilt", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)

      Set Rng9 = .Find(What:="Quarter", _
                              After:=.Cells(.Cells.Count), _
                              LookIn:=xlValues, _
                              LookAt:=xlWhole, _
                              SearchOrder:=xlByRows, _
                              SearchDirection:=xlNext, _
                              MatchCase:=False)


  End With

  'We map the ranges and the column names
  Units_col = Rng1.Column
  Init_acq_cost_col = Rng2.Column
  PropertyType_col = Rng3.Column
  CapRate_col = Rng4.Column
  Year_col = Rng5.Column
  NetSalePrice_col = Rng6.Column
  Prop_col = Rng7.Column
```

```
Year_built_col = Rng8.Column
Quarter_col = Rng9.Column

'We set up a variable with the number of rows in the initial file
Dim nb_rows As Single
nb_rows = Cells(Sheets("Original_Data").Rows.Count, 1).End(xlUp).Row

'nb_columns give the last column of the array - we define it as equal to the column of the cap rates (last column)
Dim nb_columns As Single
nb_columns = CapRate_col

'Initial array
Dim tableau() As Variant
ReDim tableau(1 To nb_rows, 1 To nb_columns)

'We first load the global range of cells in a variable called "tableau"
For i = 1 To nb_rows
    For j = 1 To nb_columns
         tableau(i, j) = Cells(i, j).Value
    Next j
Next i

'we initialise the variables which are going to count the number of times a property is verified and then redimension the arrays.
Dim elements_apartment_nb_rows() As Variant
ReDim elements_apartment_nb_rows(1 To nb_rows, 1 To nb_columns)

Dim nb_apartment_type_selected As Single
nb_apartment_type_selected = 0

'apartment
For i = 1 To nb_rows
    For j = 1 To nb_columns

        If tableau(i, PropertyType_col) = "A" Then
            elements_apartment_nb_rows(i, j) = tableau(i, j)

        Else:
            elements_apartment_nb_rows(i, PropertyType_col) = 0
        End If

    Next j

Next i

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''PROPERTY TYPE SELECTION'''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Quick Sort Array
'has been taken at:
'StackOverflow http://stackoverflow.com/questions/4873182/sorting-a-multidimensionnal-array-in-vba

'each time we filter the data through the column expressed at the end of the code line
'all the non zeros values are put in the end
Call Quick_sort.QuickSortArray(elements_apartment_nb_rows, , , PropertyType_col)

'then we just keep the variables we are interested in - when it equals 0 we remove them
'we count the number of time the initial variable "tableau" verifies a property type. This will be needed then to redimension the array
'and just keep the elements of the Property type "apartment" in this case.
For i = 1 To nb_rows

        If tableau(i, PropertyType_col) = "A" Then

            nb_apartment_type_selected = nb_apartment_type_selected + 1

```

```
204         End If
205
206 Next i
207
208 Dim elements_apartment_type_selected() As Variant
209 ReDim elements_apartment_type_selected(1 To nb_apartment_type_selected - 1, 1 To
    nb_columns)
210
211 Dim nb_apartment_type_selected_and_init_acq_cost_diff_than_zero As Single
212 nb_apartment_type_selected_and_init_acq_cost_diff_than_zero = 0
213
214 'we paste the values from one array to one another - we just want to keep the values
    which do not equal 0
215
216 ''apartment
217 For i = 1 To nb_apartment_type_selected - 1
218     For j = 1 To nb_columns
219         elements_apartment_type_selected(i, j) = elements_apartment_nb_rows(nb_rows -
            i, j)
220         'just for testing
221 '        ActiveWorkbook.Worksheets("Test1").Activate
222 '        Cells(i + 1, 1).Value =
    elements_apartment_type_selected(nb_apartment_type_selected - 1 - i, Year_col)
223     Next j
224 Next i
225
226 For i = 1 To nb_apartment_type_selected - 1
227
228         If elements_apartment_type_selected(i, Init_acq_cost_col) > 0 Then
229
230         'here we count the number of times the initial acquisition costs are equal to 0
            in order to remove them afterwards
231         nb_apartment_type_selected_and_init_acq_cost_diff_than_zero =
            nb_apartment_type_selected_and_init_acq_cost_diff_than_zero + 1
232
233         Else:
234         elements_apartment_type_selected(i, Init_acq_cost_col) = 0
235         End If
236 Next i
237
238 'then we remove the columns where we have zeros and we just keep the rows for which the
    initial cost is strictly superior to one
239 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''
240 ''''''''''''''''''''''''''''''''''''''''INITIAL ACQUISITION
    COST''''''''''''''''''''''''''''''''''''''''
241 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''''''''''
242
243 'we filter according to the column initial acquisition costs
244 Call Quick_sort.QuickSortArray(elements_apartment_type_selected, , , Init_acq_cost_col)
245
246 'we need to remove the values which equal 0 - so we need to redimension the array
247 Dim elements_apartment_type_selected_and_init_acq_cost_diff_than_zero() As Variant
248 ReDim elements_apartment_type_selected_and_init_acq_cost_diff_than_zero(1 To
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero - 1, 1 To nb_columns)
249
250 Dim nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero
    As Single
251 nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero = 0
252
253
254 'apartment
255 For i = 1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero - 1
256     For j = 1 To nb_columns
257
258         elements_apartment_type_selected_and_init_acq_cost_diff_than_zero(i, j) =
            elements_apartment_type_selected(nb_apartment_type_selected - i, j)
259
```

```
        Next j
    Next i

    For i = 1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero - 1

            If elements_apartment_type_selected_and_init_acq_cost_diff_than_zero(i, CapRate_col) > 0 Then


            nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero = nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero + 1

            Else:
                elements_apartment_type_selected_and_init_acq_cost_diff_than_zero(i, CapRate_col) = 0
            End If
    Next i


    'now we remove the cap rates which are inferior or equal to zero
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    '''''''''''''''''''''''''''''''''''''''''CAP RATES'''''''''''''''''''''''''''''''''''''''''
    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Call Quick_sort.QuickSortArray(elements_apartment_type_selected_and_init_acq_cost_diff_than_zero, , , CapRate_col)

    Dim elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero() As Variant
    ReDim elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero(1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero - 1, 1 To nb_columns)

    Dim nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_net_sales_price_sup_zero As Single
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_net_sales_price_sup_zero = 0

    'apartment
    For i = 1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero - 1
        For j = 1 To nb_columns


            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero(i, j) = elements_apartment_type_selected_and_init_acq_cost_diff_than_zero(nb_apartment_type_selected_and_init_acq_cost_diff_than_zero - i, j)
            'we count the number of zeros and then we will remove all the values which equal 0

        Next j
    Next i
    For i = 1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero - 1
            'we remove the properties which have a price equal to 1: we find it is not very realistic
            If elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero(i, NetSalePrice_col) > 0 Then

            'here we count the number of times the initial acquisition costs are equal to 0 in order to remove them afterwards
```

```
302
303
            nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zer
            o_and_net_sales_price_sup_zero =
            nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zer
            o_and_net_sales_price_sup_zero + 1
304
305         Else:
306
                elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                es_sup_zero(i, NetSalePrice_col) = 0
307         End If
308 Next i
309
310
311 'we remove the values of the sale price which equal 0 or which are inferior to 0
312 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''
313 ''''''''''''''''''''''''''''''''''''''''NET SALES
    PRICE''''''''''''''''''''''''''''''''''''''''''''
314 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''
315
316
317 Call
    Quick_sort.QuickSortArray(elements_apartment_type_selected_and_init_acq_cost_diff_than_ze
    ro_and_cap_rates_sup_zero, , , NetSalePrice_col)
318
319 Dim
    elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_
    and_net_sales_price_sup_zero() As Variant
320 ReDim
    elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_
    and_net_sales_price_sup_zero(1 To
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero - 1, 1 To nb_columns)
321
322 Dim
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero_and_initial_year_built_sup_to_zero As Single
323 nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero_and_initial_year_built_sup_to_zero = 0
324
325 'apartment
326 For i = 1 To
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero - 1
327     For j = 1 To nb_columns
328
329
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero(i, j) =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero(nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates
            _sup_zero - i, j)
330         'we count the number of zeros and then we will remove all the values which
            equal 0
331
332     Next j
333 Next i
334
335 For i = 1 To
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero - 1
336         'we just keep years which are strictly superior to 1000
337         If
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero(i, Year_built_col) > 0 Then
338
```

```
339          'here we count the number of times the initial acquisition costs are equal to 0
             in order to remove them afterwards
340
341
             nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zer
             o_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero =
             nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zer
             o_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero + 1
342
343
344          Else:
345
                 elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                 es_sup_zero_and_net_sales_price_sup_zero(i, Year_built_col) = 0
346          End If
347  Next i
348
349  '''''''''''''''TEST
350  'we print the values to see
351  'ActiveWorkbook.Worksheets("Test").Activate
352  'For i = 1 To
     nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
     t_sales_price_sup_zero - 1
353  '    Cells(i, 3).Value =
     elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_
     and_net_sales_price_sup_zero(i, NetSalePrice_col)
354  'Next i
355
356  'we remove the values of the initial year built which are empty
357  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
     ''''''
358  ''''''''''''''''''''''''''''''''''''''INITIAL YEAR
     BUILT''''''''''''''''''''''''''''''''''''''
359  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
     ''''''
360  Call
     Quick_sort.QuickSortArray(elements_apartment_type_selected_and_init_acq_cost_diff_than_ze
     ro_and_cap_rates_sup_zero_and_net_sales_price_sup_zero, , , Year_built_col)
361
362  Dim
     elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_
     and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero() As Variant
363  ReDim
     elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_
     and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(1 To
     nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
     t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1, 1 To nb_columns)
364
365  'apartment
366  For i = 1 To
     nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
     t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
367      For j = 1 To nb_columns
368
369
             elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
             up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, j) =
             elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
             up_zero_and_net_sales_price_sup_zero(nb_apartment_type_selected_and_init_acq_cost
             _diff_than_zero_and_cap_rates_sup_zero_and_net_sales_price_sup_zero - i, j)
370
371      Next j
372  Next i
373
374  'We add to the year column the quarter - it is going to be used afterwards - the values
     for the years are converted into strings and then again longs values
375  'apartment
376  For i = 1 To
     nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
```

```
    t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
377
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Year_col) =
        CStr(elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_
        sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
        Year_col))
378
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Quarter_col)
        =
        CStr(elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_
        sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
        Quarter_col))
379
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Year_col) =
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Year_col) &
        "" &
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Quarter_col)
380
        elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_z
        ero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i, Year_col) =
        CLng(elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_
        sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
        Year_col))
381 Next i
382
383 'we redimension the array and remove the values when we have zeros
384 For i = 1 To
    nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
    t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
385
386         ActiveWorkbook.Worksheets("apartment").Activate
387         Cells(i + 1, 1).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            Prop_col)
388         Cells(i + 1, 2).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            Init_acq_cost_col)
389         Cells(i + 1, 3).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            NetSalePrice_col)
390         Cells(i + 1, 4).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            Year_col)
391         Cells(i + 1, 5).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            Year_built_col)
392         Cells(i + 1, 6).Value =
            elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
            up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
            CapRate_col)
393
394 Next i
395
396 'we give headers in the sheet
397 Range("A1").Value = "Prop_col"
398 Range("B1").Value = "Init_acq_cost_col"
399 Range("C1").Value = "NetSalePrice_col"
400 Range("D1").Value = "Year_col"
401 Range("E1").Value = "Year_built_col"
```

```
Range("F1").Value = "CapRate_col"


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''GETTING UNIQUE VALUES''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''FOR THE PROPERTIES'''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'The function below has been taken from the following website:
'http://www.mrexcel.com/forum/showthread.php?649576-Extract-unique-values-from-one-column-using-VBA
'this code is then used for creating the matrix of correlation
ActiveWorkbook.Worksheets("apartment").Activate
Dim d As Object, c As Variant, lr As Long
Set d = CreateObject("Scripting.Dictionary")
lr = Cells(Rows.Count, 1).End(xlUp).Row
c = Range("A2:A" & lr)
For i = 1 To UBound(c, 1)
  d(c(i, 1)) = 1
Next i
'we paste the values - this column give us the unique property elements
ActiveWorkbook.Worksheets("Unique_Properties_apartment").Activate
Range("B2").Resize(d.Count) = Application.Transpose(d.keys)

'number of unique elements
Dim nb_unique_properties As Single
nb_unique_properties = Cells(Sheets("Unique_Properties_apartment").Rows.Count, 2).End(xlUp).Row - 1

'here we directly create the different tickers for the correlation matrix
For i = 1 To nb_unique_properties
    'Cells(1, i + 2).Value = Cells(i + 1, 2).Value
    ActiveWorkbook.Worksheets("Cap_rates_apartment").Activate
    Cells(1, i + 2).Value = ActiveWorkbook.Worksheets("Unique_Properties_apartment").Cells(i + 1, 2).Value
Next i

'we load the data of the property prices in one array
Dim unique_property_references() As Single
ReDim unique_property_references(1 To nb_unique_properties)

ActiveWorkbook.Worksheets("Cap_rates_apartment").Activate

For i = 1 To nb_unique_properties
    unique_property_references(i) = Cells(1, i + 2).Value
Next i


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''CREATION OF THE MATRIX WITH THE CAP RATES AND THE YEARS''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Then, what are we doing?
'We aim at getting a matrix with just the cap rates and the years associated so we can select them
'we select the different unique values of the properties
'we find the cap rates associated to the properties
'we say that the cap rates equal 0 if they cannot be found in the database
'this will give a very big matrix we are going to rearrange afterwards
ActiveWorkbook.Worksheets("Cap_rates_apartment").Activate
'we find these values in the database filtered

For j = 1 To nb_unique_properties
    For i = 1 To nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
```

```
462
463             If unique_property_references(j) =
                elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
                up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
                Prop_col) Then
464
465                 Cells(i + 1, j + 2) =
                    elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                    es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                    , CapRate_col)
466
467             Else:
468                 Cells(i + 1, j + 2) = 0
469             End If
470                 Cells(i + 1, 1).Value =
                    elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                    es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                    , Year_col)
471                 Cells(i + 1, 2).Value =
                    elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                    es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                    , Prop_col)
472
473         Next i
474     Next j
475
476     'then we sort the values by year for each property
477     Dim tableau_cap_rates() As Single
478     ReDim tableau_cap_rates(1 To
        nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
        t_sales_price_sup_zero_and_initial_year_built_sup_to_zero, 1 To nb_unique_properties + 2)
479
480     For i = 1 To
        nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
        t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
481
482         For j = 1 To nb_unique_properties + 2
483             'we consider all the values but the headers of the first row
484             tableau_cap_rates(i, j) = Cells(i + 1, j)
485
486         Next j
487
488     Next i
489
490     Dim tableau_cap_rates_correlation_matrix() As Single
491     ReDim tableau_cap_rates_correlation_matrix(1 To 20, 1 To nb_unique_properties + 2)
492
493     'The previous steps provided us a very big matrix. We need to rearrange it with for
        keeping just the cap rates and the years associated
494     ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''
495     ''''''''''''''''''''''''''REARRANGEMENT OF THE MATRIX WITH THE CAP RATES AND THE
        YEARS''''''''''''''''''''''''''''''''''''
496     ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        ''''''''''''''''''''''''''
497
498     For i = 1 To
        nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
        t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
499
500         For j = 1 To nb_unique_properties + 2
501
502         If tableau_cap_rates(i, 1) = "20101" And tableau_cap_rates(i, j) > 0 Then
503             tableau_cap_rates_correlation_matrix(1, j) = tableau_cap_rates(i, j)
504
505         ElseIf tableau_cap_rates(i, 1) = "20102" And tableau_cap_rates(i, j) > 0 Then
506             tableau_cap_rates_correlation_matrix(2, j) = tableau_cap_rates(i, j)
507
508         ElseIf tableau_cap_rates(i, 1) = "20103" And tableau_cap_rates(i, j) > 0 Then
```

```
            tableau_cap_rates_correlation_matrix(3, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20104" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(4, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20111" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(5, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20112" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(6, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20113" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(7, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20114" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(8, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20121" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(9, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20122" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(10, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20123" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(11, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20124" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(12, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20131" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(13, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20132" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(14, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20133" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(15, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20134" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(16, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20141" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(17, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20142" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(18, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20143" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(19, j) = tableau_cap_rates(i, j)

        ElseIf tableau_cap_rates(i, 1) = "20144" And tableau_cap_rates(i, j) > 0 Then
            tableau_cap_rates_correlation_matrix(20, j) = tableau_cap_rates(i, j)

        End If

        Next j

    Next i

    'We provide the dates in the rows in order to identify the values afterwards
    tableau_cap_rates_correlation_matrix(1, 1) = "20101"
    tableau_cap_rates_correlation_matrix(2, 1) = "20102"
    tableau_cap_rates_correlation_matrix(3, 1) = "20103"
    tableau_cap_rates_correlation_matrix(4, 1) = "20104"
    tableau_cap_rates_correlation_matrix(5, 1) = "20111"
    tableau_cap_rates_correlation_matrix(6, 1) = "20112"
    tableau_cap_rates_correlation_matrix(7, 1) = "20113"
    tableau_cap_rates_correlation_matrix(8, 1) = "20114"
    tableau_cap_rates_correlation_matrix(9, 1) = "20121"
```

```
578   tableau_cap_rates_correlation_matrix(10, 1) = "20122"
579   tableau_cap_rates_correlation_matrix(11, 1) = "20123"
580   tableau_cap_rates_correlation_matrix(12, 1) = "20124"
581   tableau_cap_rates_correlation_matrix(13, 1) = "20131"
582   tableau_cap_rates_correlation_matrix(14, 1) = "20132"
583   tableau_cap_rates_correlation_matrix(15, 1) = "20133"
584   tableau_cap_rates_correlation_matrix(16, 1) = "20134"
585   tableau_cap_rates_correlation_matrix(17, 1) = "20141"
586   tableau_cap_rates_correlation_matrix(18, 1) = "20142"
587   tableau_cap_rates_correlation_matrix(19, 1) = "20143"
588   tableau_cap_rates_correlation_matrix(20, 1) = "20144"
589
590   'we activate corr mat fi as the new sheet with the unique properties, the cap rates and
      the years
591   ActiveWorkbook.Worksheets("Cap_rates_apartment_cor_mat_fi").Activate
592
593   '20: number of times we have a potential year and quarter for this database
594   For i = 1 To 20
595       For j = 1 To nb_unique_properties
596
597       Cells(1, j + 1).Value = unique_property_references(j)
598
599       Cells(i + 1, 1) = tableau_cap_rates_correlation_matrix(i, 1)
600
601       Cells(i + 1, j) = tableau_cap_rates_correlation_matrix(i, j + 1)
602
603       Cells(i + 1, nb_unique_properties + 1) = tableau_cap_rates_correlation_matrix(i,
          nb_unique_properties + 2)
604
605       Next j
606   Next i
607
608   'we create a matrix for the equivalent of the dividends for the first type of
      simulations, i.e. the simulations where we consider that the dividends are constant
609   'how do we proceed? we take the average of all cap rates available but the one which is
      the year and the quarter of the prediction
610
611   'we first define the sums of all cap rates different then the one which is associated
      to the year and quarter of the prediction
612   Dim sum_cap_rates_without_the_most_recent_one_apartment() As Single
613   ReDim sum_cap_rates_without_the_most_recent_one_apartment(1 To nb_unique_properties)
614
615   Dim number_cap_rates_different_from_zero_apartment() As Single
616   ReDim number_cap_rates_different_from_zero_apartment(1 To nb_unique_properties)
617
618   Dim dividends_apartment() As Single
619   ReDim dividends_apartment(1 To nb_unique_properties)
620
621   'We initialise the values of the array to 0
622   For j = nb_unique_properties To 1 Step -1
623       number_cap_rates_different_from_zero_apartment(j) = 0
624   Next j
625   'we count the number of times the values are superior to 0 for each property
626   For j = nb_unique_properties To 1 Step -1
627       For i = 20 To 1 Step -1
628           If tableau_cap_rates_correlation_matrix(i, j + 2) > 0 Then
629               number_cap_rates_different_from_zero_apartment(nb_unique_properties - j +
                  1) = number_cap_rates_different_from_zero_apartment(nb_unique_properties -
                  j + 1) + 1
630           End If
631       Next i
632   Next j
633
634   'We sum all the values from the one before the last one to the first one available and
      we divide then by the "number_cap_rates_different_from_zero_apartment"
635   'This is the case when we have more than one cap rate available for each property.
      However, in some rare cases, we just have one cap rate. In this case we just
636   'the only available value as the cap rate which will be inserted later into the
      Geometric Brownian Motion.
```

```
637
638   For j = nb_unique_properties To 1 Step -1
639
640       For i = 20 To 1 Step -1
641       If tableau_cap_rates_correlation_matrix(i, j + 2) > 0 Then
642
643           If number_cap_rates_different_from_zero_apartment(nb_unique_properties - j + 1)
          > 1 Then
644               sum_cap_rates_without_the_most_recent_one_apartment(nb_unique_properties -
                  j + 1) =
                  Application.WorksheetFunction.Sum(Get_Array.getArray(Range(Cells(2, j +
                  1).Address, Cells(i, j + 1).Address)))
645               Exit For
646           Else:
647               sum_cap_rates_without_the_most_recent_one_apartment(nb_unique_properties -
                  j + 1) = tableau_cap_rates_correlation_matrix(i, j + 2)
648
649           End If
650       End If
651       Next i
652   Next j
653
654   'We now have the dividends: the division of the sum by the number of cap rates superior
      to 0 excluding the latest one
655   For j = nb_unique_properties To 1 Step -1
656       If number_cap_rates_different_from_zero_apartment(nb_unique_properties - j + 1) > 1
          Then
657       dividends_apartment(nb_unique_properties - j + 1) =
          sum_cap_rates_without_the_most_recent_one_apartment(nb_unique_properties - j + 1) /
          (number_cap_rates_different_from_zero_apartment(nb_unique_properties - j + 1) - 1)
658       Else:
659       dividends_apartment(nb_unique_properties - j + 1) =
          sum_cap_rates_without_the_most_recent_one_apartment(nb_unique_properties - j + 1) /
          number_cap_rates_different_from_zero_apartment(nb_unique_properties - j + 1)
660       End If
661   Next j
662
663
664   'we link the properties and the initial year built
665   'apartment
666   Dim apartment_initial_year_built() As Single
667   ReDim apartment_initial_year_built(1 To nb_unique_properties)
668
669   'we add the years to the simulation
670   Dim apartment_years_net_sales_prices() As Single
671   ReDim apartment_years_net_sales_prices(1 To nb_unique_properties)
672
673   'we add the prices which are going to be compared with the predictions of the
      simulations (this part is done on R)
674   Dim apartment_net_sales_prices() As Single
675   ReDim apartment_net_sales_prices(1 To nb_unique_properties)
676
677   'we add the initial_prices
678   Dim apartment_initial_prices() As Single
679   ReDim apartment_initial_prices(1 To nb_unique_properties)
680
681   'we just keep unique elements now
682   For i = 1 To
      nb_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_sup_zero_and_ne
      t_sales_price_sup_zero_and_initial_year_built_sup_to_zero - 1
683       For j = 1 To nb_unique_properties
684           If
              elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rates_s
              up_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i,
              Prop_col) = unique_property_references(j) Then
685               apartment_initial_year_built(j) =
                  elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                  es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                  , Year_built_col)
```

```
686             apartment_years_net_sales_prices(j) =
                Mid(CStr(elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_an
                d_cap_rates_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_
                to_zero(i, Year_col)), 1, 4)
687             apartment_net_sales_prices(j) =
                elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                , NetSalePrice_col)
688             apartment_initial_prices(j) =
                elements_apartment_type_selected_and_init_acq_cost_diff_than_zero_and_cap_rat
                es_sup_zero_and_net_sales_price_sup_zero_and_initial_year_built_sup_to_zero(i
                , Init_acq_cost_col)
689         End If
690     Next j
691 Next i
692
693 'we create a vector with the time differences - between initial and final prices - this
    vector will be used in the Monte Carlo simulation afterwards
694 'we initialize the variable
695 Dim apartment_time_difference() As Single
696 ReDim apartment_time_difference(1 To nb_unique_properties)
697
698 For i = 1 To nb_unique_properties
699     apartment_time_difference(i) = apartment_years_net_sales_prices(i) -
    apartment_initial_year_built(i)
700 Next i
701
702 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    ''''''
703 ''''''''''''''''''''''''''''''''''CREATION OF A NEW
    STRUCTURE''''''''''''''''''''''''''''''''''''''''
704 ''''''''''''''''''''''''''''''''''TO GATHER THE FINAL
    RESULTS''''''''''''''''''''''''''''''''''''''''
705
706
707 'We create a new structure for the apartment with just the elements we need - 'we have
    7 colomns with the important elements
708 Dim apartment_final_elements_VBA() As Single
709 ReDim apartment_final_elements_VBA(1 To nb_unique_properties, 1 To 7)
710
711 For i = 1 To nb_unique_properties
712     For j = 1 To 7
713
714         apartment_final_elements_VBA(i, 1) = unique_property_references(i)
715         apartment_final_elements_VBA(i, 2) = apartment_years_net_sales_prices(i)
716         apartment_final_elements_VBA(i, 3) = apartment_net_sales_prices(i)
717         apartment_final_elements_VBA(i, 4) = apartment_initial_year_built(i)
718         apartment_final_elements_VBA(i, 5) = apartment_initial_prices(i)
719         apartment_final_elements_VBA(i, 6) = dividends_apartment(nb_unique_properties -
            i + 1)
720         apartment_final_elements_VBA(i, 7) = apartment_time_difference(i)
721
722     Next j
723 Next i
724
725 'then with this new structure we can apply the quick sort filter to sort per years
    difference - this will be needed when using the formula of asset prices in R
726 'the seven th column corresponds to the one with the time difference
727 Call Quick_sort.QuickSortArray(apartment_final_elements_VBA, , , 7)
728
729 'we create a vector for the dividends - constant
730 ActiveWorkbook.Worksheets("apartment_final_elements_VBA").Activate
731
732 'we first declare the range where we have pasted the interest rates - we obtained them
    from https://research.stlouisfed.org/fred2/series/DTB3
733 Dim interest_rates_FED_3_m() As Single
734 '89 is the number of cells with the interest rates - for the period 1926 - 1934, as the
    data was not available, we took the same value as in 1934
```

```
ReDim interest_rates_FED_3_m(1 To 89, 1 To 2)

For i = 1 To 89
    For j = 1 To 2
     interest_rates_FED_3_m(i, j) = Cells(i + 1, 9 + j).Value
    Next j
Next i

'then we take the average for the interest rates for each property for the years considered. That means that if a price of property will be estimated
'between 1980 and 2014, we will take the average of the interest rates for this period
Dim apartment_interest_rate() As Single
ReDim apartment_interest_rate(1 To nb_unique_properties)

For j = 1 To 89
For k = 1 To 89
For i = 1 To nb_unique_properties

        If apartment_final_elements_VBA(i, 4) = interest_rates_FED_3_m(j, 1) Then

            If apartment_final_elements_VBA(i, 2) = interest_rates_FED_3_m(k, 1) Then

                apartment_interest_rate(i) = Application.WorksheetFunction.Average(Get_Array.getArray(Range("K" & (k + 1) & ":K" & (j + 1))))

        End If

    End If

Next i
Next k
Next j


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''PRINTING THE FINAL VALUES''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'We paste values
'We provide headers
Range("A1").Value = "Prop_col"
Range("B1").Value = "Year_col"
Range("C1").Value = "NetSalePrice_col"
Range("D1").Value = "Year_built_col"
Range("E1").Value = "Init_acq_cost_col"
Range("F1").Value = "CapRate_col"
Range("G1").Value = "apartment_time_difference"
Range("H1").Value = "Risk_free_rate_3_m_adapted"

'Pasting values
For i = 1 To nb_unique_properties

    For j = 1 To 7

    Cells(i + 1, j) = apartment_final_elements_VBA(i, j)

    Next j

    Cells(i + 1, 8).Value = apartment_interest_rate(i)

Next i


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''''''''''''''''''''''''''''''''''''COMING BACK TO THE CAP RATES MATRIX''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'In order to apply the R formula of asset paths, the vector time differences shall be
```

```
    ordered with a growing order.
'For this purpose, we applied the Quick Sort filter.
'However, in this case the properties are not ordered in the same way they were ordered before the application of the filter and hence,
'when computing the covariance matrix, the Properties do not correspond one to another.
'Consequently, we need to reload the matrix with the cap rates, order it with the same order as in the Sheet "apartment_final_elements_VBA" and print the values

'we activate "Cap_rates_Apartment_Final_Mtx" as the new sheet with the unique properties, the cap rates and the years
ActiveWorkbook.Worksheets("Cap_rates_Apartment_Final_Mtx").Activate

'20: number of times we have a potential year and quarter for this database

'We insert the dates
For i = 1 To 20
    Cells(i + 1, 1) = tableau_cap_rates_correlation_matrix(i, 1)
Next i

'
For j = 1 To nb_unique_properties
    For k = 1 To nb_unique_properties

          Cells(1, j + 1).Value = apartment_final_elements_VBA(j, 1)

          If apartment_final_elements_VBA(j, 1) = unique_property_references(k) Then
                For i = 1 To 20
                    Cells(i + 1, j + 1) = tableau_cap_rates_correlation_matrix(i, k + 2)
                Next i

        End If

    Next k
Next j




End Sub

```