Interface Developer Position Work

PageRank

By Simonas Šitkauskas

Contents

[Summery 3](#_Toc477857443)

[Application view: 4](#_Toc477857444)

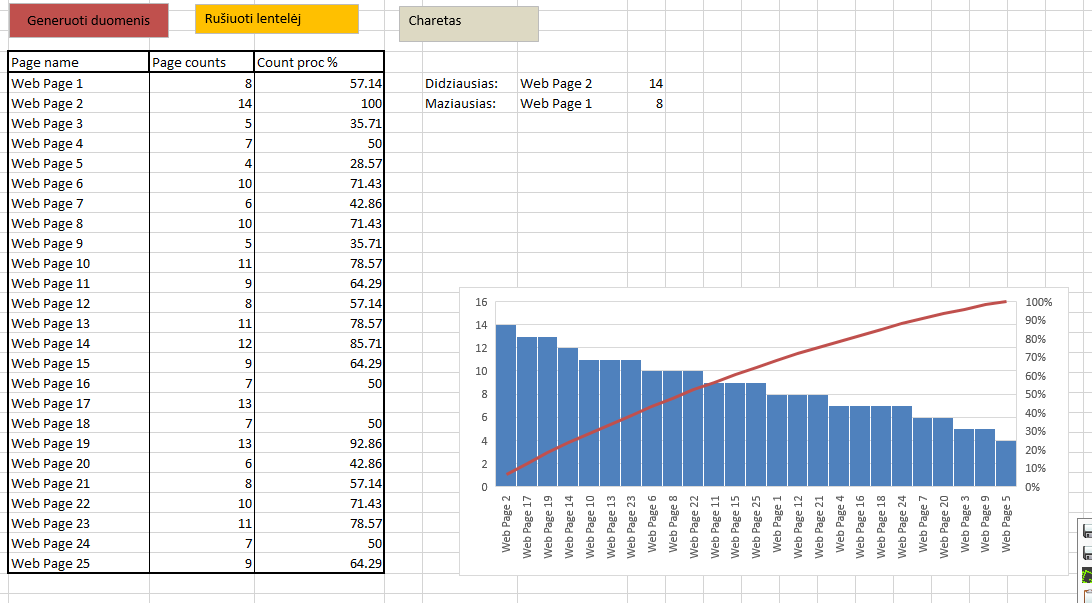
[Process 5](#_Toc477857445)

[Code 6](#_Toc477857446)

# Summery

Excel file with PageRank table, and data of pages with votes (1 or 0). Create algorithm to generate data, and count highest and lowest pages.

# Application view:



Pic. 1: View of application

This is view of application. It was created on simple excel using VBA programing langue in order to work.

# Process

It works simply by push of the buttons in excel file. Just enter data or edit in “Task Data” sheet, and go to “Sheet 1” there is 3 button you can chose. First button “Generuoti duomenis” generates data, finding biggest and lowest counts and calculating percentage then highest votes are 100% and lowest 0%. Next button “Rušiuoti lentelėje” sort table from lowest to highest and last one “Chartas” generates chart.

Sheet 1

Calculate data

Data

Generate data into array

“Generuoti duomenis”

“Rušiuoti lentelėje”

Task Data

Sort table data

**Visible table or regenerated into Sheet 1**

“Chartas”

Generate chart from table field data

**Drawn chart**

# Code

**Module1.bas / uzduotis.XLSB:** It contains VBA code, in order if import of VBA is needed to excel file. It contains all function and actions of data manipulation. Mani function is Generuoti() which generated and calculates data from “Task Data” sheet and generates new data array, for easies sorting and also max and min page with procentinge.

Sub Generuoti()

'

Sheets("Task data").Select

Dim array1(24, 1)

Dim array2(24, 2)

a = 4

minN = WorksheetFunction.Sum(Range("D4:AB4"))

maxN = WorksheetFunction.Sum(Range("D4:AB4"))

minN = 0

maxN = 0

For i = 0 To 24

array1(i, 0) = Range("C" & a)

r = "D" & a & ":AB" & a

n = WorksheetFunction.Sum(Range(r))

array1(i, 1) = n

If minN > n Then

minN = n

minI = i

End If

If maxN < n Then

maxN = n

maxI = i

End If

a = a + 1

Next

Sheets("Sheet1").Select

d = 5

For i = 0 To 24

array2(i, 0) = array1(i, 0)

array2(i, 1) = array1(i, 1)

k = 100 / (maxN - minN) \* (array1(i, 1) - minN)

array2(i, 2) = Round(k, 2)

Range("C" & d).Value = array2(i, 0)

Range("D" & d).Value = array2(i, 1)

Range("E" & d).Value = array2(i, 2)

d = d + 1

Next

Range("C4").Value = "Page name"

Range("D4").Value = "Page counts"

Range("E4").Value = "Count proc %"

Range("G5").Value = "Didziausias: "

Range("G6").Value = "Maziausias: "

Range("H5").Value = array2(maxI, 0)

Range("H6").Value = array2(minI, 0)

Range("I5").Value = array2(maxI, 1)

Range("I6").Value = array2(minI, 1)

End Sub

Sub Rusiuoti()

'

' Macro5 Macro

'

'

Sheets("Sheet1").Select

Range("C4:E29").Select

ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear

ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Range("D5:D29") \_

, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

With ActiveWorkbook.Worksheets("Sheet1").Sort

.SetRange Range("C4:E29")

.Header = xlYes

.MatchCase = False

.Orientation = xlTopToBottom

.SortMethod = xlPinYin

.Apply

End With

End Sub

Sub Macro6()

'

' Macro6 Macro

'

'

Dim rng As Range

Dim LastRow As Long

With ActiveSheet

LastRow = .Range("C29:E29").End(xlDown).Row

Set rng = Union(.Range("C4:E2").Resize(LastRow - 1), .Range("F2:F2").Resize(LastRow - 1))

End With

Charts.Add

With ActiveChart

ActiveChart.ApplyCustomType ChartType:=xlBuiltIn, TypeName:="Line - Column on 2 Axes"

.SetSourceData Source:=Range("C4:E29")

.HasTitle = True

.ChartTitle.Characters.Text = "Error"

.Axes(xlCategory, xlPrimary).HasTitle = True

.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "CS"

.Axes(xlValue, xlPrimary).HasTitle = True

.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Error#"

.Axes(xlCategory, xlSecondary).HasTitle = False

.Axes(xlValue, xlSecondary).HasTitle = True

.Axes(xlValue, xlSecondary).AxisTitle.Characters.Text = "Cum%"

.Location Where:=xlLocationAsObject, Name:="Associates"

End With

End Sub

Sub Macro7()

'

' Macro7 Macro

'

End Sub

Sub Chartt()

'This procedure is almost identical to the

'one made by the macro recorder. I have

'just changed a few things. You can make

'it more efficient, e.g. by not selecting

'objects.

Dim sSheet As String

On Error GoTo ErrorHandle

sSheet = ActiveSheet.Name

Charts.Add

ActiveChart.ChartType = xlColumnClustered

ActiveChart.SetSourceData \_

Source:=Sheets(sSheet).Range("C4:E29")

ActiveChart.Location Where:=xlLocationAsObject, Name:=sSheet

ActiveChart.SeriesCollection(2).Select

ActiveChart.SeriesCollection(2).ChartType = xlLineMarkers

ActiveChart.SeriesCollection(3).Select

With Selection.Border

.ColorIndex = 5

.Weight = xlMedium

.LineStyle = xlContinuous

End With

Selection.Shadow = False

Selection.InvertIfNegative = False

With Selection.Interior

.ColorIndex = 5

.Pattern = xlSolid

End With

ActiveChart.SeriesCollection(3).AxisGroup = 2

ActiveChart.SeriesCollection(3).ChartType = xlLine

ActiveChart.SeriesCollection(2).Select

With Selection.Border

.ColorIndex = 3

.Weight = xlMedium

.LineStyle = xlContinuous

End With

With Selection

.MarkerBackgroundColorIndex = 3

.MarkerForegroundColorIndex = 3

.MarkerStyle = xlSquare

.Smooth = False

.MarkerSize = 7

.Shadow = False

End With

ActiveChart.SeriesCollection(2).AxisGroup = 2

ActiveChart.PlotArea.Select

With Selection.Border

.ColorIndex = 16

.Weight = xlThin

.LineStyle = xlContinuous

End With

With Selection.Interior

.ColorIndex = 2

.PatternColorIndex = 1

.Pattern = xlSolid

End With

ActiveChart.Axes(xlValue, xlSecondary).Select

With ActiveChart.Axes(xlValue, xlSecondary)

.MinimumScaleIsAuto = True

.MaximumScale = 100

.MinorUnitIsAuto = True

.MajorUnitIsAuto = True

.Crosses = xlAutomatic

.ReversePlotOrder = False

.ScaleType = xlLinear

.DisplayUnit = xlNone

End With

ActiveChart.SeriesCollection(1).Select

With Selection.Border

.ColorIndex = 1

.Weight = xlThin

.LineStyle = xlContinuous

End With

Selection.Shadow = False

Selection.InvertIfNegative = False

With Selection.Interior

.ColorIndex = 15

.Pattern = xlSolid

End With

With ActiveChart.ChartGroups(1)

.Overlap = 0

.GapWidth = 0

.HasSeriesLines = False

.VaryByCategories = False

End With

With ActiveChart

.HasTitle = True

.ChartTitle.Characters.Text = "Titel"

End With

Exit Sub

ErrorHandle:

End Sub