

## Автоматизация создания панельных диаграмм в Excel

Панельные диаграммы идеально подходят для отдельного отображения сразу нескольких рядов данных. Ниже показан линейный график, явно перегруженный линиями, а справа – его аналог в виде панельной диаграммы, на которой ряды данных располагаются отдельно.

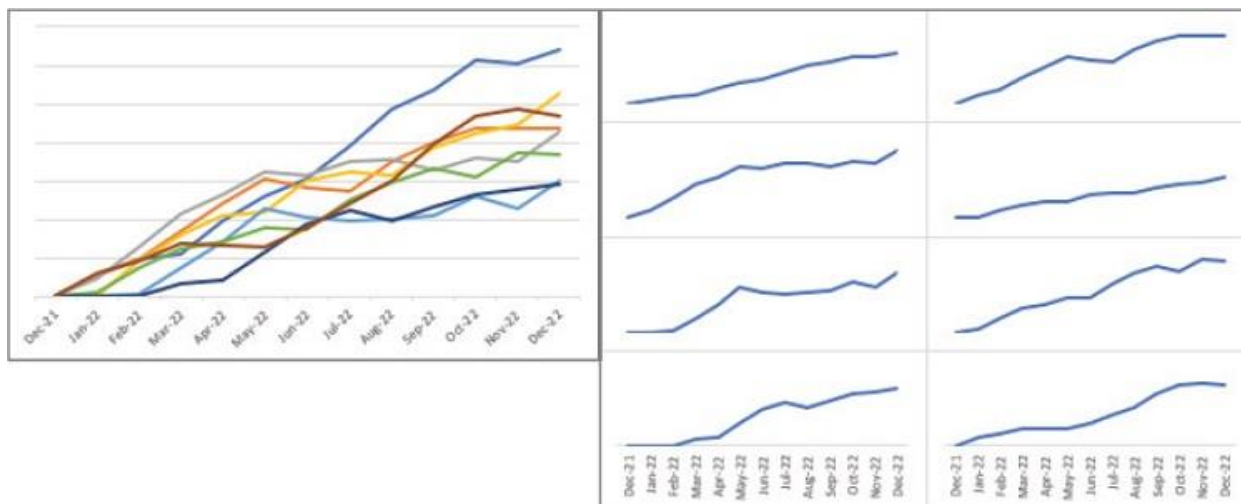


Рис. 1. Линейный график и панельная диаграмма

### Запись макроса

Размещение каждого графика на панельной диаграмме вручную – задача не из легких.<sup>1</sup> Написание макроса, который это сделает, у вас уйдет примерно столько же времени, но этот вариант имеет свои преимущества. Во-первых, с использованием макроса вы можете быть уверены, что все графики будут размещены на панели идеально четко и ни один из них не сместится ни на пиксель. Второе преимущество связано с возможными изменениями, которые могут произойти в будущем. Если вы что-то меняете на графиках, вам придется заново вручную расставлять их, на что потребуется столько же времени, сколько и в первый раз. Что касается макроса, то вам необходимо будет лишь изменить пару цифр и запустить его.

Давайте начнем с записи макроса, чтобы подсмотреть манипуляции с объектами. Определите максимальное значение в диапазоне B1:I14. Оно нам понадобится, чтобы установить шкалу ординат.  $\text{МАКС}(B2:I14)=6420$ . На вкладке *Разработчик* нажмите на кнопку *Запись макроса* и в открывшемся диалоговом окне нажмите *Ок*. Выделите диапазон A1:B14 на рабочем листе. На вкладке *Вставка* нажмите *Рекомендуемые диаграммы* и выберите вариант *График*. Удалите горизонтальные линии сетки. Выделите вертикальную ось, задайте максимальное значение 7000 и оформите ее, как показано на рис. 2. Измените размер диаграммы. Переместите диаграмму. Не важно, куда – нам лишь нужно получить соответствующий код. Оформите горизонтальную ось. На вкладке *Разработчик* нажмите *Остановить запись*. В результате мы получим код макроса, показанный ниже. Чтобы посмотреть его, перейдите на вкладку *Разработчик* и нажмите на кнопку *Visual Basic*:

```
Sub Макрос1()  
'  
' Макрос1 Макрос  
'  
Range("A1:B14").Select  
ActiveSheet.Shapes.AddChart2(227, xlLineMarkers).Select  
ActiveChart.SetSourceData Source:=Range("'Рис. 2'!$A$1:$B$14")  
ActiveChart.Axes(xlValue).MajorGridlines.Select  
Selection.Delete  
ActiveSheet.ChartObjects("Диаграмма 22").Activate  
ActiveChart.Axes(xlValue).Select  
ActiveChart.Axes(xlValue).MaximumScale = 7000
```

<sup>1</sup> Написано по мотивам книги Дик Куслейка. [Визуализация данных при помощи дашбордов и отчетов в Excel](#).

```

Selection.MajorTickMark = xlOutside
With Selection.Format.Line
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorBackground1
    .ForeColor.TintAndShade = 0
    .ForeColor.Brightness = -0.150000006
End With
ActiveChart.Axes(xlCategory).Select
ActiveChart.Axes(xlCategory).MajorUnit = 3
ActiveChart.ChartArea.Select
ActiveSheet.Shapes("Диаграмма 22").IncrementLeft -376.5
ActiveSheet.Shapes("Диаграмма 22").IncrementTop 44.25
ActiveSheet.Shapes("Диаграмма 22").ScaleWidth 1.1677084427, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes("Диаграмма 22").ScaleHeight 1.04340296, msoFalse, _
    msoScaleFromTopLeft
End Sub

```

	A	B	C	D	E	F	G	H	I
1	Date	Series1	Series2	Series3	Series4	Series5	Series6	Series7	Series8
2	Dec-21	0	0	0	0	0	0	0	0
3	Jan-22	533	555	454	0	0	115	0	596
4	Feb-22	980	947	1265	935	57	727	0	903
5	Mar-22	1121	1684	2124	1592	726	1225	330	1383
6	Apr-22	1951	2400	2641	2107	1413	1408	423	1309
7	May-22	2594	3071	3246	2170	2282	1760	1137	1268
8	Jun-22	3050	2810	3127	3022	2046	1724	1848	1775
9	Jul-22	3932	2711	3487	3241	1972	2517	2212	2422
10	Aug-22	4861	3514	3553	3160	2009	2982	1978	2990
11	Sep-22	5364	4003	3278	3886	2102	3332	2312	3964
12	Oct-22	6118	4383	3576	4242	2587	3085	2647	4666
13	Nov-22	6067	4384	3524	4465	2289	3750	2797	4880
14	Dec-22	6420	4371	4266	5263	3000	3667	2934	4701

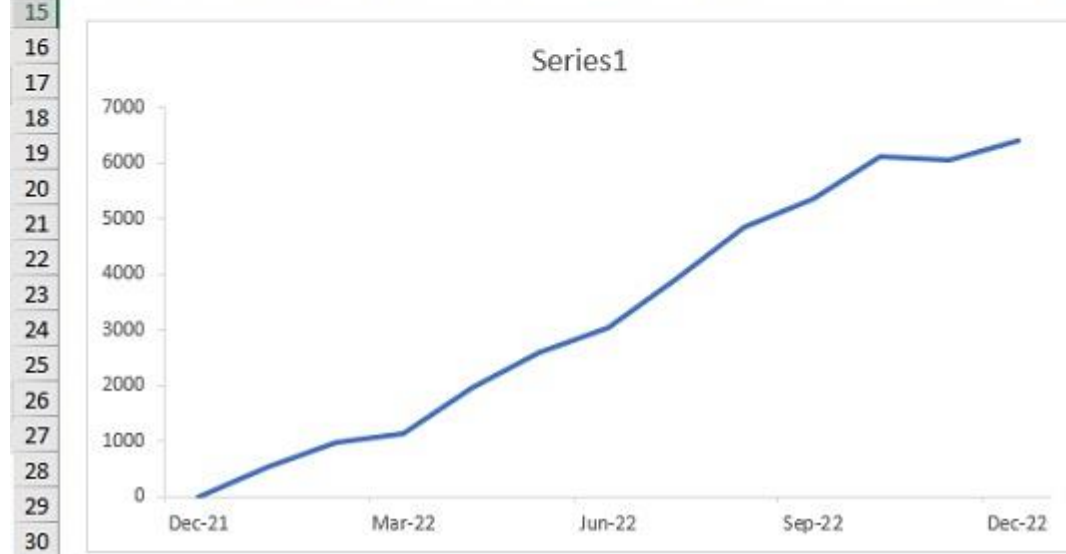


Рис. 2. Запись макроса

При записи макроса Excel записывает строки кода для всех действий. Так, первая строка создана в ответ на выделение диапазона ячеек, вторая – на создание диаграммы и т.д. Запись макросов полезно использовать для изучения объектов модели Excel и синтаксиса выражений. В данном случае вы видите, что нам придется поработать с объектами и методами AddChart2, SetSourceData, ActiveChart и ChartObjects. Не беспокойтесь, если не понимаете сгенерированный

код, мы будем использовать его лишь как образец. И повторно запустить этот макрос у вас не получится, так как Excel создаст новую диаграмму с названием отличным от "Диаграмма 22".

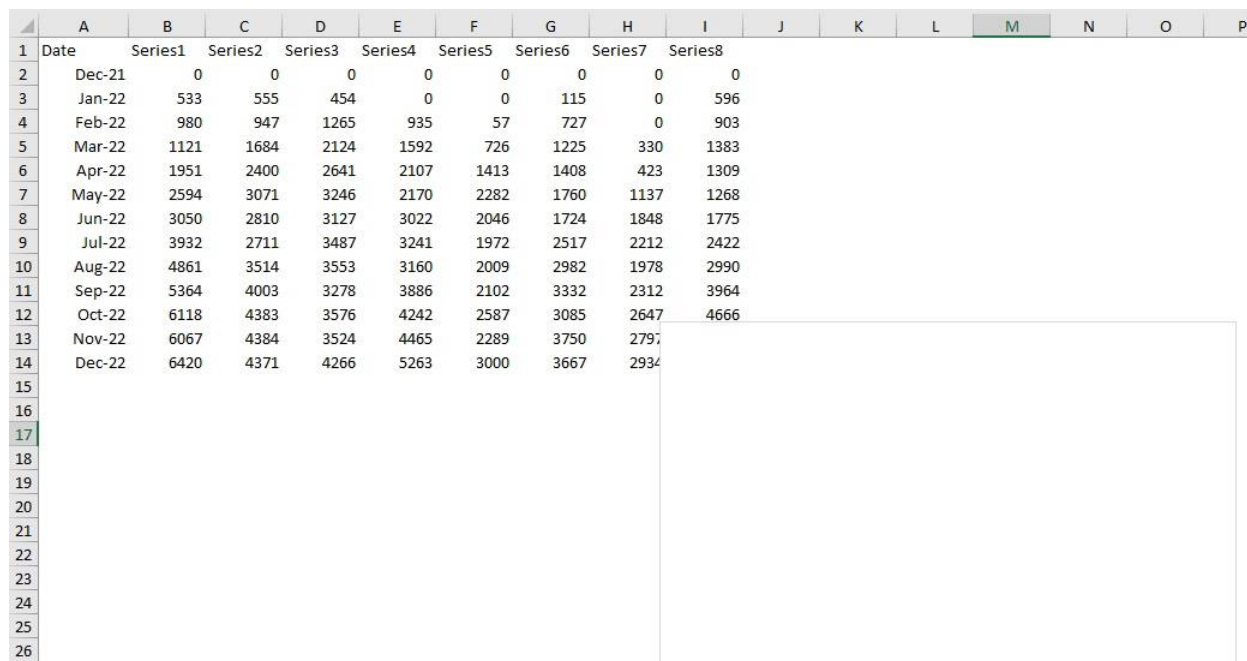
Вы, наверное, уже обратили внимание, что во время ваших действий Excel выбирает или активирует те или иные объекты, а затем выполняет какие-то действия с ними. В макросе делается только так, поскольку Excel не знает заранее, будете ли вы выполнять какие-то действия с объектом при его выборе. Но в своем коде вам не стоит предварительно выбирать объекты для работы.

### Создание графика

Ниже кода макроса введите Sub MakeSinglePane и нажмите Enter. VBA автоматически добавит скобки к названию подпрограммы и завершит ее инструкцией End Sub. Ключевые слова Sub и End Sub ограничивают начало и окончание вашей программы. Создадим переменную, в которой будем хранить диаграмму. Вместо использования объекта ActiveChart мы присвоим созданный график переменной и будем манипулировать с ней. Введите следующий код между ключевыми словами Sub и End Sub:

```
Sub MakeSinglePanel()  
    Dim cht As Chart  
    Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart  
End Sub
```

При помощи ключевого слова Dim мы создаем переменную типа график, а выражение Set позволяет присвоить ей конкретный тип графика. Константа xlLine гарантирует скрытие маркеров на графике. Параметр 227 задает стиль диаграммы. Если вы запустите этот макрос, то получите пустую диаграмму на рабочем листе.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Date	Series1	Series2	Series3	Series4	Series5	Series6	Series7	Series8							
2	Dec-21	0	0	0	0	0	0	0	0							
3	Jan-22	533	555	454	0	0	115	0	596							
4	Feb-22	980	947	1265	935	57	727	0	903							
5	Mar-22	1121	1684	2124	1592	726	1225	330	1383							
6	Apr-22	1951	2400	2641	2107	1413	1408	423	1309							
7	May-22	2594	3071	3246	2170	2282	1760	1137	1268							
8	Jun-22	3050	2810	3127	3022	2046	1724	1848	1775							
9	Jul-22	3932	2711	3487	3241	1972	2517	2212	2422							
10	Aug-22	4861	3514	3553	3160	2009	2982	1978	2990							
11	Sep-22	5364	4003	3278	3886	2102	3332	2312	3964							
12	Oct-22	6118	4383	3576	4242	2587	3085	2647	4666							
13	Nov-22	6067	4384	3524	4465	2289	3750	2797								
14	Dec-22	6420	4371	4266	5263	3000	3667	2934								
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																

Рис. 3. Пустая диаграмма

Нам же необходимо манипулировать переменной cht для добавления элементов на график.

### График с данными

Начнем с установки диапазона данных для диаграммы. Для этого используем связку With и End With для экономии чернил и лучшей организации кода. Добавьте в ваш макрос:

```
Sub MakeSinglePanel()  
    Dim cht As Chart  
    Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart  
    With cht  
        .SetSourceData ActiveSheet.Range("A1:B14")  
    End With  
End Sub
```

Если запустить этот код, мы получим график с данными:

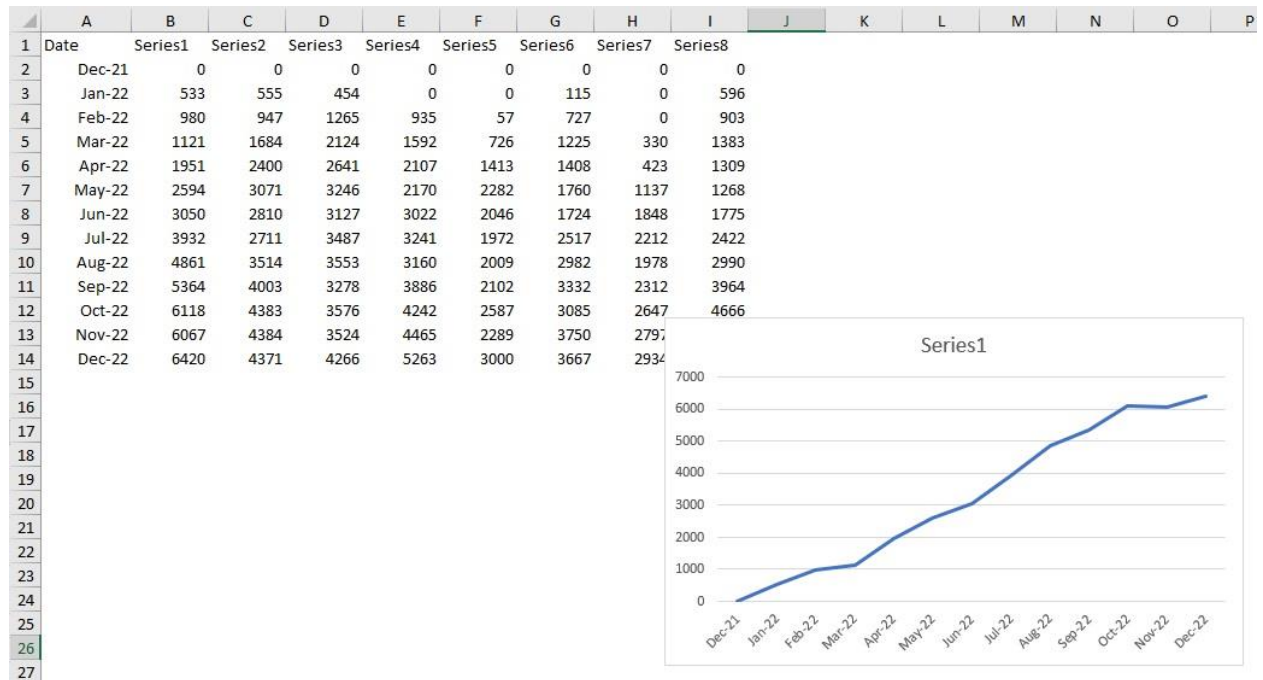


Рис. 4. Диаграмма с исходными данными

### Форматирование графика

Отформатируем ось ординат, ось абсцисс, удалим горизонтальные линии сетки:

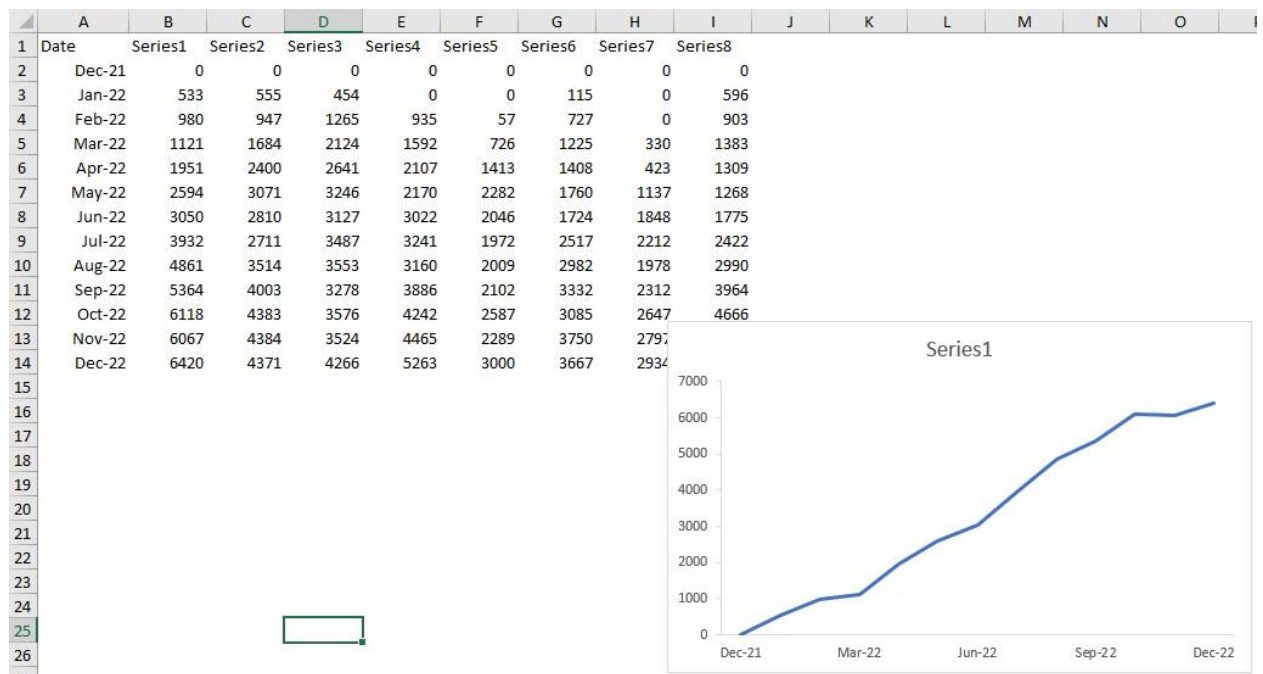


Рис. 5. Отформатированы оси

### Sub MakeSinglePanel()

Dim cht As Chart

Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart

With cht

.SetSourceData ActiveSheet.Range("A1:B14")

.Axes(xlValue).MajorGridlines.Delete

.Axes(xlValue).MinimumScale = 0

.Axes(xlValue).MaximumScale = 7000

.Axes(xlValue).MajorTickMark = xlOutside

With .Axes(xlValue).Format.Line

.Visible = msoTrue

```

        .ForeColor.ObjectThemeColor = msoThemeColorBackground1
        .ForeColor.TintAndShade = 0
        .ForeColor.Brightness = -0.150000006
    End With
    .Axes(xlCategory).MajorUnit = 3
End With
End Sub

```

### Позиционирование графика

Изменим размер графика и переместим его в нужное место. Автоматически записанный макрос оперирует для этого объектом `Shape` с использованием свойств и методов для изменения масштаба объекта и перемещения относительно текущей позиции. Я не знаю, почему Excel делает именно так, но для манипуляций есть более простой способ. До этого мы манипулировали объектом `Chart`, у которого есть родительский объект `ChartObject`. И у этого родительского объекта присутствуют свойства вроде `Top` и `Height`, позволяющие устанавливать координаты элемента напрямую. В следующем фрагменте кода мы добавим еще один блок `With` и используем свойство `Parent` нашего графика, чтобы изменить размеры и положение нашего объекта:

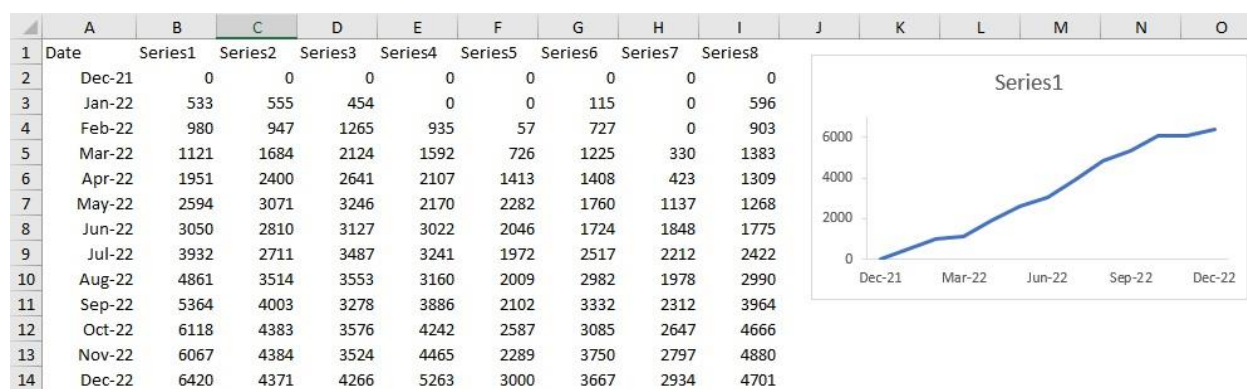


Рис. 6. Маленькая диаграмма, как элемент панельной инфографики

```

Sub MakeSinglePanel()
    Dim cht As Chart
    Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart
    With cht
        .SetSourceData ActiveSheet.Range("A1:B14")
        .Axes(xlValue).MajorGridlines.Delete
        .Axes(xlValue).MinimumScale = 0
        .Axes(xlValue).MaximumScale = 7000
        .Axes(xlValue).MajorTickMark = xlOutside
    With .Axes(xlValue).Format.Line
        .Visible = msoTrue
        .ForeColor.ObjectThemeColor = msoThemeColorBackground1
        .ForeColor.TintAndShade = 0
        .ForeColor.Brightness = -0.150000006
    End With
        .Axes(xlCategory).MajorUnit = 3
    With .Parent
        .Top = 10
        .Left = 460
        .Height = 145
        .Width = 260
    End With
    End With
End Sub

```

Теперь наш код в точности повторяет действия записанного макроса. Вы можете запускать его многократно, и создаваемые графики будут размещаться один поверх другого.

### *Создание панели из восьми графиков*

В наши же планы входит создание восьми графиков, а не одного. Вы можете использовать для этого наш макрос MakeSinglePanel, но для каждого графика вам придется вручную менять его координаты. Лучше создать макрос MakeSinglePanel2, который будет принимать на вход аргументы и использовать их в коде:

```
Sub MakeSinglePanel2(rSource As Range, _
    dTop As Double, dLeft As Double, _
    dHeight As Double, dWidth As Double)
Dim cht As Chart
Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart
With cht
    .SetSourceData rSource
    .Axes(xlValue).MajorGridlines.Delete
    .Axes(xlValue).MinimumScale = 0
    .Axes(xlValue).MaximumScale = 7000
    .Axes(xlValue).MajorTickMark = xlOutside
    With .Axes(xlValue).Format.Line
        .Visible = msoTrue
        .ForeColor.ObjectThemeColor = msoThemeColorBackground1
        .ForeColor.TintAndShade = 0
        .ForeColor.Brightness = -0.150000006
    End With
    .Axes(xlCategory).MajorUnit = 3
    With .Parent
        .Top = dTop
        .Left = dLeft
        .Height = dHeight
        .Width = dWidth
    End With
End With
End Sub
```

При создании первого макроса вручную VBA автоматически добавлял скобки после названия подпрограммы. Внутри этих скобок вы можете задавать аргументы, которые будут поступать на вход макроса. Аргументы внутри макроса работают как обычные переменные, за исключением того, что их значения не задаются в макросе, а поступают извне при вызове подпрограммы. В данном случае мы добавили аргумент типа Range, с помощью которого будем задавать исходный диапазон для графика, а также четыре аргумента типа Double, которые помогут нам позиционировать диаграмму. Вы можете вызвать подпрограмму MakeSinglePanel2 из другого макроса, передав ей на вход необходимые аргументы. Например:

```
MakeSinglePanel2 ActiveSheet.Range("A1:B14"), 10, 460, 145, 260
```

Таким образом, наши переменные не инициализируются внутри макроса, а их значения поступают извне. Вы можете передать в макрос разные аргументы, что позволит построить разные диаграммы. Давайте вызовем наш макрос восемь раз с четко выверенными значениями аргументов для создания восьми графиков в рамках единой панели:

```
Sub MakeAllPanels()
    With ActiveSheet
        MakeSinglePanel2 .Range("A1:B14"), 10, 460, 145, 260
        MakeSinglePanel2 .Range("A1:A14, C1:C14"), 155, 460, 145, 260
        MakeSinglePanel2 .Range("A1:A14, D1:D14"), 300, 460, 145, 260
        MakeSinglePanel2 .Range("A1:A14, E1:E14"), 445, 460, 145, 260
        MakeSinglePanel2 .Range("A1:A14, F1:F14"), 10, 720, 145, 260
        MakeSinglePanel2 .Range("A1:A14, G1:G14"), 155, 720, 145, 260
        MakeSinglePanel2 .Range("A1:A14, H1:H14"), 300, 720, 145, 260
        MakeSinglePanel2 .Range("A1:A14, I1:I14"), 445, 720, 145, 260
    End With
End Sub
```

End With  
End Sub

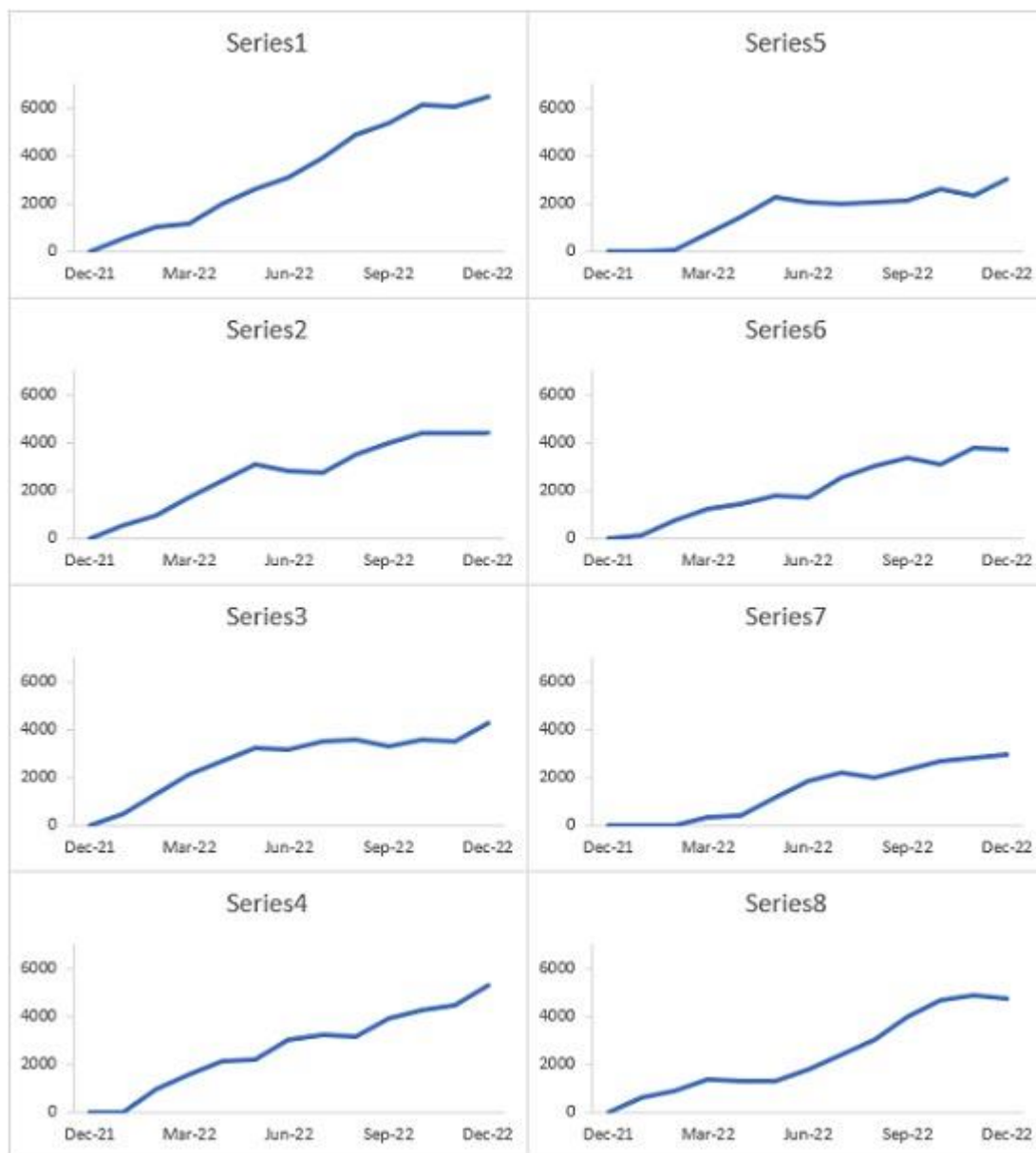


Рис. 7. Панель из 8 графиков

### *Организация цикла*

Возможно, вы заметили закономерности в переданных макросу MakeSinglePanel2 аргументах. Всегда, когда вы замечаете определенные шаблоны в коде, у вас должно возникать желание реализовать выполнение операции в цикле, чтобы добавить макросу гибкости. Давайте создадим новую подпрограмму MakeAllPanels2(), в которой создание и размещение наших диаграмм на панели будет выполняться в цикле. Это позволит в будущем легко менять размеры элементов и их расположение. В новом макросе для разнообразия разместим диаграммы сначала слева направо, а затем сверху вниз:

```
Sub MakeAllPanels2()  
    Dim rAxis As Range  
    Dim i As Long, j As Long  
    Dim lCnt As Long  
    Dim dWidth As Double, dHeight As Double  
    Set rAxis = ActiveSheet.Range("A1:A14")  
    dTop = 10  
    dLeft = 460  
    dWidth = 260
```

```
dHeight = 145
```

```
For i = 1 To 4
  For j = 1 To 2
    lCnt = lCnt + 1
    MakeSinglePanel2 _
      rSource:=Union(rAxis, rAxis.Offset(0, lCnt)), _
      dTop:=dTop + ((i - 1) * dHeight), _
      dLeft:=dLeft + ((j - 1) * dWidth), _
      dHeight:=dHeight, _
      dWidth:=dWidth
  Next j
Next i
End Sub
```

Вся основная работа в этом макросе выполняется внутри двух вложенных циклов For. В первом определяется высота панели, а во втором – ширина. Поскольку мы используем один диапазон A1:A14 для оси абсцисс, мы заранее сохранили его в переменной. Затем мы использовали выражение Union для объединения этого диапазона с нужной нам колонкой со значениями по оси ординат.

Верхнюю левую координату графиков мы отсчитываем от значений 10 и 460. Каждый раз при запуске итерации внешнего цикла мы добавляем к предыдущей координате по вертикали высоту элемента, чтобы новый график разместился точно под предыдущим. То же самое происходит с горизонтальным смещением во внутреннем цикле. Если вы захотите разместить диаграммы в формате 4\*2, то можете просто объявить внешний цикл как For i = 1 to 2, а внутренний – как For j = 1 to 4.

#### *Оформление панели, как единого целого*

Наличие горизонтальной оси на всех графиках не является обязательным. Оставим подписи только на двух нижних графиках. Проблема с удалением некоторых подписей состоит в том, что размер диаграмм с удаленной осью автоматически увеличится, что нарушит наши пропорции. Чтобы это обойти, мы можем зафиксировать значение свойства PlotArea.InsideHeight для всех графиков. Если устанавливать свойство PlotArea.InsideHeight напрямую, все ваши графики будут масштабированы одинаково, но значение свойства ChartArea.Height не изменится, и в местах, где были оси, останется пустое пространство.

Напишем новый макрос MakeSinglePanel3. На этот раз это будет не подпрограмма, а функция. Функция отличается от подпрограммы тем, что может возвращать значение или объект в вызывающий блок кода. Наша функция будет возвращать созданный график, в котором свойство PlotArea.InsideHeight можно будет установить позже. Функция также содержит два новых аргумента: dInsideHeight и bHideAxis:

```
Function MakeSinglePanel3(rSource As Range, _
  dTop As Double, dLeft As Double, _
  dHeight As Double, dWidth As Double, _
  dInsideHeight As Double, bHideAxis As Boolean) As Chart

  Dim i As Long
  Dim cht As Chart

  Set cht = ActiveSheet.Shapes.AddChart2(227, xlLine).Chart

  With cht
    .SetSourceData rSource
    .Axes(xlValue).MajorGridlines.Delete
    .Axes(xlValue).MinimumScale = 0
    .Axes(xlValue).MaximumScale = 7000
    .Axes(xlValue).MajorTickMark = xlOutside
```



```

With .Axes(xlValue).Format.Line
    .Visible = msoTrue
    .ForeColor.ObjectThemeColor = msoThemeColorBackground1
    .ForeColor.TintAndShade = 0
    .ForeColor.Brightness = -0.150000006
End With
.Axes(xlCategory).MajorUnit = 3
With .Parent
    .Top = dTop
    .Left = dLeft
    .Height = dHeight
    .Width = dWidth
End With
If bHideAxis Then
    With .Axes(xlPrimary)
        .Delete
        .HasMajorGridlines = False
    End With
    .Parent.Height = dHeight
Else
    .Parent.Height = dHeight
    Do Until .PlotArea.InsideHeight > dInsideHeight
        .Parent.Height = .Parent.Height + 1
    Loop
End If
End With

```

```
Set MakeSinglePanel3 = cht
```

```
End Function
```

Аргумент bHideAxis отвечает за то, будет ли скрыта ось на графике. Если он равен True, основная ось будет скрыта. В противном случае мы имеем дело с диаграммой в нижнем ряду панели, и высота будет увеличиваться, пока область построения не сравняется по размерам с другими графиками. В заключительной строке созданный график возвращается в вызывающий код путем присваивания его имени функции.

Вызывающую подпрограмму также потребуется изменить. Нам придется определять, находится ли график на нижнем ряду панели, чтобы правильно устанавливать значение аргумента bHideAxis.

```
Sub MakeAllPanels3()
```

```

Dim rAxis As Range
Dim i As Long, j As Long
Dim lCnt As Long
Dim dWidth As Double, dHeight As Double
Dim dInsideHeight As Double
Dim cht As Chart

```

```

Const lHigh As Long = 4
Const lWide As Long = 2

```

```
Set rAxis = ActiveSheet.Range("A1:A14")
```

```

dTop = 10
dLeft = 460
dWidth = 230
dHeight = 130

```

```

For i = 1 To IHigh
  For j = 1 To IWide

    ICnt = ICnt + 1
    Set cht = MakeSinglePanel3( _
      rSource:=Union(rAxis, rAxis.Offset(0, ICnt)), _
      dTop:=dTop + ((i - 1) * dHeight), _
      dLeft:=dLeft + ((j - 1) * dWidth), _
      dHeight:=dHeight, _
      dWidth:=dWidth, _
      dInsideHeight:=dInsideHeight, _
      bHideAxis:=i < IHigh)

    If i = 1 And j = 1 Then
      dInsideHeight = cht.PlotArea.InsideHeight
    End If
  Next j
Next i

End Sub

```

Эта подпрограмма содержит две новые переменные для хранения высоты области построения и объекта с графиком, возвращаемого функцией. Также мы объявили две константы с высотой и шириной панели. Вместо того чтобы вставлять значения непосредственно в циклы For Next, вы можете объявить константы в самом начале процедуры и использовать их при необходимости. Таким образом, если вы захотите изменить ориентацию панели, вам необходимо будет обновить значения констант и все.

Здесь мы вызываем не подпрограмму для создания графиков, а функцию, возвращающую объект диаграммы, который мы присваиваем переменной cht. При таком вызове функции все аргументы должны быть заключены в круглые скобки. После вызова функции мы сохраняем значение свойства PlotArea.InsideHeight первого созданного графика, чтобы при достижении нижнего ряда функция знала, какой высоты создавать диаграммы. Итоговая панельная диаграмма:

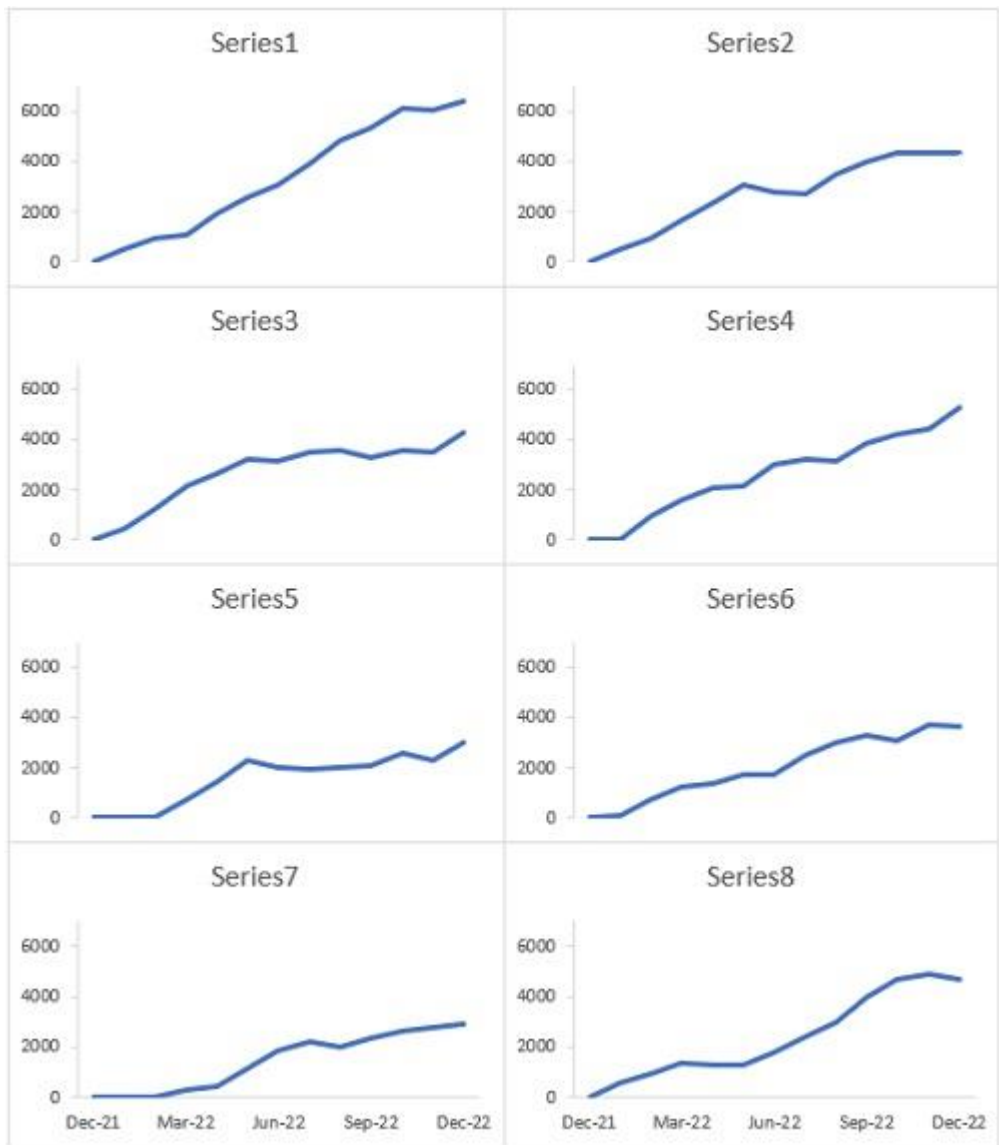


Рис. 8. Инфографика на основе панельной диаграммы