

# 教程 - 图表表达式

Qlik Sense®

May 2023

版权所有 © 1993-2023 QlikTech International AB。保留所有权利。





---

<b>1 欢迎学习本教程！</b>	<b>4</b>
1.1 您将学到的内容	4
1.2 谁应当完成该教程	4
1.3 本教程中的课程	4
1.4 延伸阅读和资源	4
<b>2 在可视化中使用表达式</b>	<b>5</b>
2.1 什么是表达式？	5
2.2 我在哪里可以使用表达式？	5
2.3 什么时候计算表达式？	5
<b>3 什么是聚合函数？</b>	<b>6</b>
3.1 使用 Sum() 合并金额	6
3.2 使用 Max() 计算最高销售值	7
3.3 使用 Min() 计算最低销售值	8
3.4 使用 Count() 计算实体的数目。	8
Count() 和 Count(distinct) 之间的差异	9
<b>4 嵌套聚合函数</b>	<b>11</b>
4.1 总是函数中的一个聚合级别	11
4.2 将 Aggr() 用于嵌套聚合函数	11
4.3 计算最大的平均订单价值	12
<b>5 裸字段引用</b>	<b>15</b>
5.1 务必在表达式中使用聚合函数	15
使用 If() 函数拆分发票日期	15
5.2 避免裸字段引用	16
在 If() 函数中避免裸字段引用	16
<b>6 The importance of Only()</b>	<b>19</b>
6.1 使用 Only() 的不同表达式	20
<b>7 现实中的例子</b>	<b>25</b>
7.1 计算总利润值百分比	25
7.2 开发票延迟	27
7.3 谢谢！	31

# 1 欢迎学习本教程！

本教程简介 Qlik Sense 中的图表表达式。表达式是函数、字段和数学运算符的组合，用于处理数据并生成可在可视化中看到的结果。

图表表达式在度量中使用最多。您还可以创建更有活力更强大的可视化，只需使用标题、副标题、脚注和维度的表达式。

## 1.1 您将学到的内容

完成本教程后，您将会掌握如何在可视化中使用表达式。

## 1.2 谁应当完成该教程

您应当熟悉 Qlik Sense 的基础知识。例如，您已加载过数据、创建过应用程序并在不同工作表上创建过可视化。

您将需要数据加载编辑器的访问权限，并且应该被允许在 Qlik Sense Enterprise on Windows 中加载数据。

## 1.3 本教程中的课程

本教程中的主题可按任何顺序完成。但是，稍后的主题将假定您熟悉之前的主题。截屏取自 Qlik Sense Enterprise SaaS。如果您在不同的部署上使用 Qlik Sense Enterprise，可能会看到一些差异。

## 1.4 延伸阅读和资源

- [Qlik](#) 提供了各种各样的资源帮助您进行深入学习。
- [Qlik 在线帮助](#) 可供使用。
- 培训，包括免费的在线课程，可在 [Qlik Continuous Classroom](#) 获取。
- 讨论论坛、博客等可见于 [Qlik Community](#)。

## 2 在可视化中使用表达式

使用图表可创建 Qlik Sense 中的可视化, 这些图表是使用维度和度量而创建。通过表达式, 您可让自己的可视化更加动态和复杂。

可视化包含标题、副标题、脚注以及帮助传递信息的其他元素。构成可视化的元素可以很简单。例如: 由表示数据的字段组成的维度, 以及由文本组成的标题。

度量是基于字段的计算。例如: **Sum(Cost)** 意味着字段 **Cost** 的所有值都使用函数 **Sum** 聚合。换言之, **Sum(Cost)** 是一个表达式。

### 2.1 什么是表达式?

表达式是函数、字段和数学运算符 (+ \* / =) 的组合。表达式用于处理应用程序中的数据, 以便生成可以在可视化中看到的结果。这些表达式可以简单(仅涉及基本计算), 也可以复杂(涉及函数字段和运算符)。表达式可用于脚本和图表可视化。

所有度量都是表达式。度量和表达式之间的差别是, 表达式没有名称或描述数据。

您可以创建更有活力更强大的可视化, 只需使用维度、标题、副标题和脚注的表达式。这表示(例如)可视化标题不是静态文本, 而是可以使用表达式产生的内容, 该表达式的结果将根据做出的选择改变。

### 2.2 我在哪里可以使用表达式?

在编辑可视化时, 如果可以在“属性”面板中看到 **fx** 符号, 则可以使用表达式。单击 **fx** 可以打开表达式编辑器, 此编辑器旨在帮助您创建和编辑表达式。也可以在表达式字段中直接输入表达式。

表达式不能直接保存为主项。但是, 主度量和主维度可以包含表达式。如果表达式用于随后保存为主条目的度量值或维度, 则保留度量值或维度中的表达式。

### 2.3 什么时候计算表达式?

在加载脚本中, 将在执行脚本之后评估表达式。在可视化中, 只要表达式包含的任何字段, 变量或函数的值或逻辑状态更改, 就会自动评估表达式。脚本表达式和图表表达式在语法和可用函数方面有一些不同。

## 3 什么是聚合函数？

聚合函数是多对一函数。它们使用来自多个记录的值作为输入，并将这些值转换为一个值来汇总所有记录。`Sum()`、`Count()`、`Avg()`、`Min()`、和 `Only()` 都是聚合函数。

在 Qlik Sense 中，您在大多数公式中刚好需要一个聚合函数级别。这包括图表表达式、文本框和标签。如果不在表达式中加入聚合函数，则 Qlik Sense 将自动分配 `Only()` 函数。

- 聚合函数是返回一个值的函数，该值描述您数据中多个记录的某些属性。
- 所有表达式(计算维度除外)都作为聚合计算。
- 表达式中的所有字段引用都必须包含在聚合函数中。



您可使用表达式编辑器来创建和更改 Qlik Sense 中的表达式。

### 3.1 使用 `Sum()` 合并金额

`Sum()` 用于计算聚合数据之间表达式或字段指定值的总和。

让我们计算每个经理的总销售额，以及所有经理的总销售额。

在应用程序内，在 *Which Aggregations?* 工作表上您将找到表格，一个表格标题为 `Sum()`、`Max()`、`Min()`，一个表格标题为 `Count()`。我们将使用每个表来创建聚合函数。

执行以下操作：

1. 选择可用的 `Sum()`、`Max()`、`Min()` 表格。  
属性面板打开。
2. 单击**添加列**并选择**度量**。
3. 单击 ***fx*** 符号。  
表达式编辑器打开。
4. 输入以下内容：`Sum(Sales)`
5. 单击**应用**。

示出每个 *Manager* 总销售额的表格

Sum(), Max (), Min()	
Manager	Sum(Sales)
<b>Totals</b>	<b>\$ 104,852,674.81</b>
Dennis Johnson	\$ 15,945,030.85
Stewart Wind	\$ 15,422,448.79
Carolyn Halmon	\$ 11,363,424.41
John Greg	\$ 9,770,909.24
Samantha Allen	\$ 7,540,947.33
Amanda Honda	\$ 6,436,630.86
Brenda Gibson	\$ 6,215,872.87
Kathy Clinton	\$ 5,154,950.48
Molly McKenzie	\$ 5,079,387.55
John Davis	\$ 4,060,007.10

您可以看到每个经理的销售额，以及所有经理的总销售额。



作为最佳实践，请确保您的数据格式正确。在该情况下，将**数字格式**设置为**货币**，并将**格式样式**设置为**\$ #,##0;-\$ #,##0**。

## 3.2 使用 Max() 计算最高销售值

**Max()** 用于查找聚合数据每行的最高值。

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容：**Max (Sales)**
4. 单击**应用**。

示出每个经理总销售额和最高销售额的表格

Sum(), Max (), Min()		
Manager	Sum(Sales)	Max(Sales)
<b>Totals</b>	<b>\$ 104,852,674.81</b>	<b>\$ 555,376.00</b>
Dennis Johnson	\$ 15,945,030.85	\$ 285,350.40
Stewart Wind	\$ 15,422,448.79	\$ 258,946.70
Carolyn Halmon	\$ 11,363,424.41	\$ 555,376.00
John Greg	\$ 9,770,909.24	\$ 310,156.07
Samantha Allen	\$ 7,540,947.33	\$ 52,469.65
Amanda Honda	\$ 6,436,630.86	\$ 133,568.68
Brenda Gibson	\$ 6,215,872.87	\$ 119,030.00
Kathy Clinton	\$ 5,154,950.48	\$ 47,326.42
Molly McKenzie	\$ 5,079,387.55	\$ 79,134.97
John Davis	\$ 4,060,007.10	\$ 118,210.17

您可以看到每个经理的最高销售收益，以及最高总数。

### 3.3 使用 Min() 计算最低销售值

**Min()** 用于查找聚合数据每行的最低值。

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 ***fx*** 符号。  
表达式编辑器打开。
3. 输入以下内容：*Min (Sales)*
4. 单击**应用**。

示出每个经理总销售额、最高销售额以及最低销售额的表格

Sum(), Max (), Min()			
Manager	Sum(Sales)	Max(Sales)	Min(Sales)
<b>Totals</b>	<b>\$ 104,852,674.81</b>	<b>\$ 555,376.00</b>	<b>-\$ 27,929.88</b>
Dennis Johnson	\$ 15,945,030.85	\$ 285,350.40	-\$ 27,929.88
Stewart Wind	\$ 15,422,448.79	\$ 258,946.70	-\$ 1,687.63
Carolyn Halmon	\$ 11,363,424.41	\$ 555,376.00	-\$ 13,749.60
John Greg	\$ 9,770,909.24	\$ 310,156.07	-\$ 17,883.07
Samantha Allen	\$ 7,540,947.33	\$ 52,469.65	-\$ 1,687.91
Amanda Honda	\$ 6,436,630.86	\$ 133,568.68	-\$ 15,122.77
Brenda Gibson	\$ 6,215,872.87	\$ 119,030.00	-\$ 11,903.00
Kathy Clinton	\$ 5,154,950.48	\$ 47,326.42	-\$ 3,418.90
Molly McKenzie	\$ 5,079,387.55	\$ 79,134.97	-\$ 1,631.49
John Davis	\$ 4,060,007.40	\$ 110,240.47	-\$ 12,770.70

您可以看到每个经理的最低销售收益，以及最低总数。

### 3.4 使用 Count() 计算实体的数目。

**Count()** 用于计算每个图表维度中值、文本和数字的数量。

在我们的数据中，每个经理对许多销售代表 (*Sales Rep Name*) 负责。我们来计算销售代表的数目。

执行以下操作：

1. 选择可用的 **Count()** 表格。  
属性面板打开。
2. 单击**添加列**并选择**度量**。
3. 单击 ***fx*** 符号。  
表达式编辑器打开。
4. 输入以下内容：*Count([Sales Rep Name])*
5. 单击**应用**。

示出销售代表和销售代表总数的表格



Count()		
Sales Rep Name	Q	Count([Sales Rep Name])
Totals		64
Amalia Craig		1
Amanda Honda		1
Carl Lynch		1
Molly McKenzie		1
Sheila Hein		1
Brenda Gibson		1
Dennis Johnson		1
Ken Roberts		1
Robert Kim		1
William Fisher		1
Cary Frank		1
Edward Smith		1
Lee Chin		1
Ronald Milam		1

可以看到销售代表的总数是 64。

#### Count() 和 Count(distinct) 之间的差异

我们来计算经理的数目。

执行以下操作：

1. 向您的表格新增维度：*Manager*。  
单个经理处理多个销售代表，因此同一个经理名称在表中出现多次。
2. 单击**添加列**并选择**度量**。
3. 单击 ***fx*** 符号。  
表达式编辑器打开。
4. 输入以下内容：*Count(Manager)*
5. 用表达式另外添加一个度量：*Count(distinct Manager)*
6. 单击**应用**。

以表格显示销售代表、销售代表总数、每个 *Sales Representative* 的负责 *Manager*、不正确的 *Manager* 总数和正确的 *Manager* 总数。

### 3 什么是聚合函数？

Count()				
Sales Rep Name	Count([Sales Rep Name])	Manager	Count(Manager)	Count(distinct Manager)
Totals	64		64	18
Amalia Craig	1	Amanda Honda	1	1
Amanda Honda	1	Amanda Honda	1	1
Carl Lynch	1	Amanda Honda	1	1
Molly McKenzie	1	Amanda Honda	1	1
Sheila Hein	1	Amanda Honda	1	1
Brenda Gibson	1	Brenda Gibson	1	1
Dennis Johnson	1	Brenda Gibson	1	1
Ken Roberts	1	Brenda Gibson	1	1
Robert Kim	1	Brenda Gibson	1	1
William Fisher	1	Brenda Gibson	1	1
Cary Frank	1	Carolyn Halmon	1	1
Edward Smith	1	Carolyn Halmon	1	1
Lee Chin	1	Carolyn Halmon	1	1
Ronald Milam	1	Carolyn Halmon	1	1

您可以看到使用 *Count(Manager)* 作为表达式的列上经理的总数计算为 64。这是不正确的。使用 *Count(distinct Manager)* 表达式, 经理总数正确地计算为 18。每个经理只计入一次, 而不考虑他们的姓名在列表中出现的次数。

## 4 嵌套聚合函数

图表表达式中的任何字段名称必须正好被一个聚合函数包围。如果需要嵌套聚合，可以使用 **Aggr()** 添加第二聚合级别。**Aggr()** 包含聚合函数作为参数。

### 4.1 总是函数中的一个聚合级别

典型的应用程序可包含：

- 数据中的一百万个记录
- 透视表中的一百行
- 表盘或文本框中的单个 KPI

尽管量级不同，所有三个数字仍然表示所有数据。数字仅为不同的聚合级别。

聚合函数使用来自多个记录的值作为输入，并将这些值转换为一个单独的值，该值可视为所有记录的摘要。存在一个限制：不能在另一个聚合函数中使用聚合函数。您通常需要将每个字段引用包含在确切的一个聚合函数中。

以下表达式将有效：

- **Sum(Sales)**
- **Sum(Sales) / Count(Order Number)**

以下表达式将不起作用，因为它是嵌套聚合：

- **Count(Sum(Sales))**

这个问题的解决方法是 **Aggr()** 形式的函数。与其名称相反，它不是聚合函数。它是一个“多对多”函数，更像是数学中的矩阵。其将具有 **N** 个记录的表格转换成具有 **M** 个记录的表格。其返回值的数组。它也可以被看作具有一个度量单位和一个或多个维度的虚拟垂直表。



如果想要创建多层次嵌套图表聚合，则在计算维度中使用 **Aggr()** 函数。

### 4.2 将 **Aggr()** 用于嵌套聚合函数

**Aggr()** 用于返回在声明维度或维度上计算的表达式的值的阵列。例如，每个区域的每位客户的最大销售额值。在高级聚合中，其中在另一个聚合函数中要将 **Aggr()** 函数括起来，从而将 **Aggr()** 函数的结果阵列用作其所嵌套的聚合的输入。

当使用它时，**Aggr()** 语句生成一个虚拟表，其中一个表达式按一个或多个维度分组。然后可通过外部聚合函数进一步聚合该虚拟表的结果。

## 4.3 计算最大的平均订单价值

让我们在图表表达式中使用简单的 **Aggr()** 语句。

我们希望在区域级别看到我们的总体指标，但也要显示两个更复杂的表达式：

- 每个区域内按经理划分的最大平均订单价值。
- 对该最大平均订单价值负责的经理。

我们可以使用标准表达式 **Sum(Sales)/Count([Order Number])** 方便地计算每个地区的平均订单价值。

在应用程序内，在 *Nested Aggregations* 工作表上您将找到标题为 *Aggr() function* 的表格。

执行以下操作：

1. 选择可用的 **Aggr() function** 表格。  
属性面板打开。
2. 单击**添加列**并选择**度量**。
3. 单击 **fx** 符号。  
表达式编辑器打开。
4. 输入以下内容：**Sum(Sales)/Count([Order Number])**
5. 单击**应用**。

以表格示出每个地区的平均订单价值。

Aggr() function	
Region	Average order value
<b>Totals</b>	<b>\$ 1,887</b>
Germany	\$ 405
Japan	\$ 604
Nordic	\$ 641
Spain	\$ 577
UK	\$ 1,390
USA	\$ 1,821



作为最佳实践，请确保您的数据格式正确。在该情况下，在每个列中我们将更改**标签**以表示计算。在具有货币值的列中，我们将**数字格式**更改为**货币**，并将**格式样式**更改为 \$ #,##0;- \$ #,##0。

我们的目标是检索每个地区的最大平均订单价值。我们必须使用 **Aggr()** 告知 Qlik Sense，我们希望获取每个地区每个经理的平均订单值，然后显示其中最大的一个。要获得每个地区、每个经理的平均订单值，我们必须在 **Aggr()** 语句中包括以下维度：

**Aggr (Sum (Sales) / Count ([Order Number]), Region, Manager)**

该表达式导致 Qlik Sense 生成如下所示的虚拟表

示出每个地区每个经理的平均订单价值的 **Aggr()** 函数的虚拟表。

Virtual table of Aggr() function

Region	Manager	Average order value
<b>Totals</b>		-
Germany	Micheal Williams	\$ 3,506
Germany	Dennis Johnson	\$ 1,380
Germany	Molly McKenzie	\$ 820
Germany	David Laychak	\$ 624
Germany	John Davis	\$ 456
Germany	Sheila Hein	\$ 445
Germany	Amanda Honda	\$ 443
Germany	John Greg	\$ 436
Germany	Samantha Allen	\$ 404
Germany	Stewart Wind	\$ 393
Germany	William Fisher	\$ 380
Germany	Ken Roberts	\$ 379
Germany	Kathy Clinton	\$ 335
Germany	Odessa Morris	\$ 331

当 Qlik Sense 计算每个地区每个经理的单个平均订单值时，我们需要找到这些值中最大的一个。我们通过用 **Max()** 环绕 **Aggr()** 函数来进行该操作。

**Max (Aggr (Sum (Sales) / Count ([Order Number]), Manager, Region))**

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容：**Max(Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))**
4. 单击**应用**。

以表格示出地区、每个地区每个经理的平均订单价值以及最大平均订单价值。

Aggr() function		
Region	Average order value	Largest average order value
<b>Totals</b>	<b>\$ 1,087</b>	<b>\$ 12,338</b>
Germany	\$ 405	\$ 3,506
Japan	\$ 604	\$ 2,182
Nordic	\$ 641	\$ 2,554
Spain	\$ 577	\$ 1,639
UK	\$ 1,390	\$ 12,338
USA	\$ 1,821	\$ 8,615

您可以看到所有经理在地区级别的最大平均订单值。这是我们两个复杂表达式中的第一个！下一个要求是让负责这些较大平均订单价值的经理的姓名自行显示在这些值旁边。

为此，我们将如前使用同一 **Aggr()** 函数，但是这次将连同 **FirstSortedValue()** 函数一起使用。

**FirstSortedValue()** 函数告知 Qlik Sense 针对函数第二部分中指定的特定维度为我们提供经理：

**FirstSortedValue (Manager, Aggr (Sum (Sales) / Count (Order Number), Manager, Region))**



表达式有一个很小但很重要的部分：**Aggr()** 表达式前面的减号。在 **FirstSortedValue()** 函数内，您可指定数据数组的排序顺序。在该情况下，减号符号告知 Qlik Sense 从最大到最小排序。

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 ***fx*** 符号。  
表达式编辑器打开。
3. 输入以下内容：**FirstSortedValue(Manager,-Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))**
4. 单击**应用**。

以表格示出地区、每个地区的平均订单价值和最大平均订单价值，以及负责该订单价值的经理。

Aggr() function				
Region	Q	Average order value	Largest average order value	Manager
Totals		\$ 1,087	\$ 12,338	Dennis Johnson
Germany		\$ 405	\$ 3,506	Micheal Williams
Japan		\$ 604	\$ 2,182	Brenda Gibson
Nordic		\$ 641	\$ 2,554	Kathy Clinton
Spain		\$ 577	\$ 1,639	Micheal Williams
UK		\$ 1,390	\$ 12,338	Dennis Johnson
USA		\$ 1,821	\$ 8,615	Carolyn Halmon

# 5 裸字段引用

如果字段未包含在聚合函数中，则将其视为裸字段。

裸字段引用是一个数组，可能包含多个值。如果是这样，Qlik Sense 将把其评估为 NULL，不知道您需要这些值中的哪些。

## 5.1 务必在表达式中使用聚合函数

如果发现表达式的计算不正确，则很有可能它没有聚合函数。

表达式中的字段引用是值的数组。例如：

两个表格，一个示出 **Max(Invoice Date)** 为单个值，一个示出 *Invoice Date* 为值数组。

Max(Invoice Date)	Invoice Date
Max([Invoice Date])	Invoice Date
6/26/2014	1/12/2012
	1/13/2012
	1/18/2012
	1/19/2012
	1/20/2012
	1/21/2012
	1/22/2012
	1/25/2012
	1/26/2012

必须将字段 *发票日期* 括在聚合函数中，使其转变为单个值。

如果不对表达式使用聚合函数，则默认情况下 Qlik Sense 将使用 **Only()** 函数。如果字段引用返回多个值，Qlik Sense 将把它解释为 NULL。

### 使用 **If()** 函数拆分发票日期

**If()** 函数通常用于条件聚合。它返回一个值，具体取决于函数内提供的条件的计算结果是 True 还是 False。

在应用程序内，在 *Naked field references* 工作表上您将找到标题为 *Using If() on Invoice dates* 的表格。

执行以下操作：

1. 选择标题为 *Using If() on Invoice dates* 的可用表格。  
属性面板打开。
2. 单击 **添加列** 并选择 **度量**。
3. 单击 **fx** 符号。  
表达式编辑器打开。
4. 输入以下内容：*If ([Invoice Date]>= Date(41323), 'After', 'Before')*
5. 单击 **应用**。

在表格中按参考日期分割示出发票日期。

Using If() on Invoice dates		
Date	Q	if([Invoice Date]>= Date(41323), 'After', 'Before')
Totals		Before
2/10/2013		Before
2/11/2013		Before
2/12/2013		Before
2/13/2013		Before
2/14/2013		Before
2/17/2013		Before
2/18/2013		After
2/19/2013		After
2/20/2013		After
2/21/2013		After
2/24/2013		After
2/25/2013		After

该表达式测试 *Invoice Date* 是否在参考日期 2013 年 2 月 18 日之前, 如果是则返回 'Before'。如果日期晚于或等同于参考日期 2013 年 2 月 18 日, 则会返回 'After'。参考日期表示为整数 41323。

## 5.2 避免裸字段引用

初看之下, 该表达式没什么问题。

```
If([Invoice Date]>= Date(41323) 'After', 'Before')
```

它应当计算引用日期之后的发票日期, 返回 'After' 或返回 'Before'。但是, *Invoice Date* 是裸字段引用, 没有聚合函数, 因此是具有数个值的数组, 并且将计算为 NULL。在上一示例中, 我们表格中每个 *Date* 值仅有一个 *Invoice Date*, 因此表达式的计算没什么问题。

让我们看看类似的表达式如何在不同的维度值下计算, 以及如何解决裸字段引用问题:

### 在 If() 函数中避免裸字段引用

我们将和之前一样使用相似的表达式:

```
If([Invoice Date]>= Date(41323), Sum(Sales))
```

这一次, 函数对参考日期之后的销售额求和。

在应用程序内, 在 *Naked field references* 工作表上您将找到标题为 *Sum(Amount)* 的表格。

执行以下操作:

1. 选择可用的 *Sum(Amount)* 表格。  
属性面板打开。
2. 单击 **添加列** 并选择 **度量**。
3. 单击 **fx** 符号。  
表达式编辑器打开。
4. 输入以下内容: *If([Invoice Date]>= 41323, Sum(Sales))*
5. 单击 **应用**。

以表格使用 **If()** 函数示出年份、每年销售额总和以及表达式结果。



Sum(Amount)		
Year	Sum(Sales)	If([Invoice Date]>= Date(41323), Sum(Sales))
Totals	\$ 104,852,675	-
2012	\$ 40,173,302	-
2013	\$ 42,753,991	-
2014	\$ 21,925,382	-



在度量上保持**标签**完好，以示出每个表达式之间的差异。在具有货币值的列中，将**数字格式**设置为**货币**，并将**格式样式**设置为\$ #,##0;- \$ #,##0。

对于每一年，都有一个在参考日期之后的发票日期数组。由于我们的表达式缺少聚合函数，其将计算为 NULL。正确的表达式应当在 **If()** 函数的第一参数中使用聚合函数，诸如 **Min()** 或 **Max()**：

```
If (Max ([Invoice Date]) >= Date (41323), Sum (Sales) )
```

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容：If([Invoice Date]>= Date(41323), Sum(Sales) )
4. 单击**应用**。

以表格使用 **If()** 函数示出年份、每年销售额总和以及不同表达式结果。

Sum(Amount)			
Year	Sum(Sales)	If([Invoice Date]>= Date(41323), Sum(Sales))	If(Max([Invoice Date])>= Date(41323), Sum(Sales))
Totals	\$ 104,852,675	-	\$ 104,852,675
2012	\$ 40,173,302	-	-
2013	\$ 42,753,991	-	\$ 42,753,991
2014	\$ 21,925,382	-	\$ 21,925,382

或者，可以将 **If()** 函数置于 **Sum()** 函数内：

```
Sum (If ([Invoice Date]>= Date (41323), Sales) )
```

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容：Sum( If([Invoice Date]>= Date(41323), Sales) )
4. 单击**应用**。

以表格使用 **If()** 函数示出年份、每年销售额总和以及不同表达式结果。

Sum(Amount)				
Year	Sum(Sales)	If( [Invoice Date]>= Date(41323), Sum(Sales) )	If(Max([Invoice Date])>= Date(41323), Sum(Sales) )	Sum( If([Invoice Date]>= Date(41323), Sales))
Totals	\$ 104,852,675	-	\$ 104,852,675	\$ 58,563,348
2012	\$ 40,173,302	-	-	\$ 0
2013	\$ 42,753,991	-	\$ 42,753,991	\$ 36,637,967
2014	\$ 21,925,382	-	\$ 21,925,382	\$ 21,925,382

在第二个到最后一个表达式中, **If()** 函数按每个维度值计算一次。在最后一个表达式中, 它在原始数据中每行计算一次。函数运算方式的不同导致结果不同, 但两者都返回一个答案。第一个表达式直接运算为 NULL。上图显示了两个表达式之间的差异, 使用 2013 年 2 月 18 日作为参考日期。

## 6 The importance of Only()

如果组中只有一个可能值, **Only()** 会返回一个值。该值将为聚合的结果。如果未指定聚合函数, 则 Qlik Sense 默认设置为 **Only()**。

如果图表维度和参数之间存在一对一关系, 则 **Only()** 函数返回唯一可能的值。如果存在数个值, 则会返回 **NULL**。例如, 如果有多个产品的单价为 12, 则只搜索单价为 12 的产品将会返回 **NULL**。

下图显示了一对一和一对多关系之间的区别:

一个表格显示 *Manager Number* 和 *Manager* 之间的一对一关系

One-to-one relationship	
Manager Number	Manager
104	Amanda Honda
109	Brenda Gibson
111	Carolyn Halmon
118	David Laychak
121	Dennis Johnson
132	John Davis
134	John Greg
144	Kathy Clinton
145	Ken Roberts
157	Micheal Williams
159	Molly McKenzie
160	Odessa Morris
169	Samantha Allen
176	Sheila Hein
179	Stephanie Reagan
181	Stewart Wind
184	Viginia Mountain
185	William Fisher

一个表格显示 *Sales Rep Name* 和 *Manager* 的一对多关系

One-to-many relationship	
Sales Rep Name	Manager
Amalia Craig	Amanda Honda
Amanda Honda	Amanda Honda
Cart Lynch	Amanda Honda
Molly McKenzie	Amanda Honda
Sheila Hein	Amanda Honda
Brenda Gibson	Brenda Gibson
Dennis Johnson	Brenda Gibson
Ken Roberts	Brenda Gibson
Robert Kim	Brenda Gibson
William Fisher	Brenda Gibson
Cary Frank	Carolyn Halmon
Edward Smith	Carolyn Halmon
Lee Chin	Carolyn Halmon
Ronald Milam	Carolyn Halmon
Amelia Fields	David Laychak
Deborah Halmon	David Laychak
Judy Rowlett	David Laychak
Angelen Carter	Dennis Johnson
Dennis Fisher	Dennis Johnson

**Only()** 函数是一种聚合函数。它使用许多记录作为输入，只返回一个值，类似于 **Sum()** 或 **Count()**。Qlik Sense 几乎会在其所有计算中使用聚合。图表、排序表达式、文本框、高级搜索和计算标签中的表达式都是聚合，如果不涉及聚合函数，则无法计算。

但是，如果用户输入的表达式缺少显式聚合函数会怎样？例如，如果排序表达式设置为 *Date* 会怎样？或者，如果有一个针对购买过啤酒和葡萄酒产品的客户并使用表达式 `= [Product Type] = 'Beer and Wine'` 的高级搜索，又会怎么样？

这就是 **Only()** 函数影响计算的地方。如果表达式中没有显式聚合函数，则 Qlik Sense 隐式使用 **Only()** 函数。在上述情况中，**Only(Date)** 用作排序表达式，而 **Only([Product Type]) = 'Beer and Wine'** 用作搜索条件。

有时新表达式返回用户不期望的结果。当只有一个 *Date* 或 *Product Type* 的可能值时，上述两个示例都适用，但当有多个值时，这两个示例都不适用。

## 6.1 使用 Only() 的不同表达式

我们将用相似表达式创建四个 KPI。这样，我们可以比较在表达式中使用裸字段引用或不同 **Only()** 位置使用裸字段引用对选择结果的影响。

在应用程序内的 *Importance of Only()* 工作表上，您将找到以 *Invoice Date* 作为维度的筛选器窗格。

执行以下操作：

1. 创建 KPI。
2. 单击**添加度量项**。单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容：`Month([Invoice Date])`
4. 使用以下度量另外创建三个 KPIs：`Month(Only([Invoice Date]))`、`Month(Max([Invoice Date]))` 和 `Only(Month([Invoice Date]))`。
5. 单击**应用**。

四个 KPIs 以及筛选器窗格显示三个不同但是相似的表达式。

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Jun	Only(Month([Invoice Date])) —
<div>Q Invoice Date</div> <div> <div>1/12/2012</div> <div>1/13/2012</div> <div>1/18/2012</div> <div>1/19/2012</div> <div>1/20/2012</div> <div>1/21/2012</div> <div>1/22/2012</div> </div>	



在每个 KPI 中，**数字格式**已设置为**度量表达式**。

当有裸字段引用时，将在最低级别插入 **Only()** 函数。这意味着前两个 KPI、`Month([Invoice Date])` 和 `Month(Only([Invoice Date]))` 将被解释为相同的，并始终给出相同的结果。

您可以看到四个 KPI 中有三个返回了 NULL。第三个 KPI、`Month(Max([Invoice Date]))` 已返回值，即使没有进行选择。

编写表达式时，应始终确定要使用哪个聚合，或者如果有多个值，则要使用哪个值。如果要使用 NULL 来表示多个值，可以将表达式保持原样。对于数字，您可能需要改为使用 **Sum()**、**Avg()**、**Min()** 或 **Max()**。对于字符串您可能希望使用 **Only()** 或 **MinString()**。

执行以下操作：

1. 停止编辑工作表。
2. 在“筛选器”窗格中，选择一月中的日期。
3. 通过单击 ✓ 确认选择。

KPI 结果在进行单个选择时更改。

Month([Invoice Date]) Jan	Month(Only([Invoice Date])) Jan
Month(Max([Invoice Date])) Jan	Only(Month([Invoice Date])) Jan

Q Invoice Date

1/12/2012 ✓
1/13/2012
1/18/2012
1/19/2012
1/20/2012
1/21/2012
1/22/2012

但进行单个选择时，所有 KPI 返回正确答案。即使表达式包含裸字段引用(如 `Month([Invoice Date])` 中的表达式)，我们进行了唯一选择这一事实也可让它返回正确的值。

执行以下操作：

1. 在“筛选器”窗格中，选择一月中的其他日期。
2. 通过单击 ✓ 确认选择。

当两次选择的日期都在 1 月份时，KPI 结果会发生变化。

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Jan	Only(Month([Invoice Date])) Jan
Q Invoice Date	
1/12/2012 ✓	
1/13/2012 ✓	
1/18/2012	
1/19/2012	
1/20/2012	
1/21/2012	
1/22/2012	

前两个 KPI 返回 NULLL, 其他两个 KPI 返回正确的一月的值。具体来说, 第四个 KPI 返回正确的答案, 因为我们选择的两个日期都是一月份的日期。

执行以下操作:

1. 在“筛选器”窗格中, 选择一月之外月份中的其他日期。
2. 通过单击 ✓ 确认选择。

当用不同月份的日期进行多个选择时, KPI 结果会发生变化。

Month([Invoice Date]) —	Month(Only([Invoice Date])) —
Month(Max([Invoice Date])) Feb	Only(Month([Invoice Date])) —

Q Invoice Date

1/12/2012 ✓

1/13/2012 ✓

2/1/2012 ✓

1/18/2012

1/19/2012

1/20/2012

1/21/2012

当用不同月份的日期进行多个选择时，只有第三个 KPI 返回值。根据表达式 `Month(Max([Invoice Date]))`，它返回所做选择中最大月份的值。由于 `Only()` 是自动插入到具有裸字段引用的表达式中，因此不能总是假设最低级别将适合您的表达式这点。`Only()` 的放置很重要。



## 7 现实中的例子

Qlik Sense 中的可视化可为您提供您数据中的见解。在您的图表中使用表达式可产生特别适用于您的工作的结果。Qlik Sense 中函数的范围可让您自定义自己的表达式以满足自己的需求,即使选项尚不可用。

### 7.1 计算总利润值百分比

---

我们将利润值定义为我们的销售额和进行这些销售的成本之间的差值。我们将计算每个月的利润值,以及每月销售额的多少百分比是我们的利润值。

要计算利润百分比,我们可以使用以下表达式:

```
(Sum(Sales) - Sum(Cost)) / Sum(Sales)
```

可将表达式进一步简化

```
1 - Sum(Cost) / Sum(Sales)
```

在应用程序内,在 *Examples from real life* 工作表上您将找到标题为 *Margin* 的表格。

执行以下操作:

1. 选择标题为 *Margin* 的可用表格。  
属性面板打开。
2. 单击**添加列**并选择**度量**。
3. 单击 ***fx*** 符号。  
表达式编辑器打开。
4. 输入以下内容: *Sum(Sales)*
5. 使用表达式另外添加三个度量: *Sum(Cost)*、*Sum(Sales) - Sum(Cost)* 和 *1 - Sum(Cost)/Sum(Sales)*。
6. 单击**应用**。

表格示出销售额的总和,以及每个月成本的总和,以及以金额和百分比形式表示的计算的每月利润值

Margin					
Month	Q	Sum(Sales)	Sum(Cost)	Calculated Margin	Margin %
Totals		\$ 104,852,675	\$ 61,571,565	\$ 43,281,110	41%
2012-Jan		\$ 1,773,750	\$ 1,122,474	\$ 651,276	37%
2012-Feb		\$ 3,867,568	\$ 2,352,955	\$ 1,514,613	39%
2012-Mar		\$ 3,892,195	\$ 2,339,154	\$ 1,553,041	40%
2012-Apr		\$ 3,660,634	\$ 2,241,036	\$ 1,419,598	39%
2012-May		\$ 3,191,648	\$ 1,961,629	\$ 1,230,019	39%
2012-Jun		\$ 4,259,260	\$ 2,540,976	\$ 1,718,284	40%
2012-Jul		\$ 2,519,873	\$ 1,488,274	\$ 1,031,598	41%
2012-Aug		\$ 3,799,274	\$ 2,312,303	\$ 1,486,971	39%
2012-Sep		\$ 3,739,098	\$ 2,239,469	\$ 1,499,629	40%
2012-Oct		\$ 3,036,456	\$ 1,897,354	\$ 1,139,102	38%
2012-Nov		\$ 3,528,099	\$ 2,193,961	\$ 1,334,138	38%
2012-Dec		\$ 2,905,449	\$ 1,693,359	\$ 1,212,089	42%
2013-Jan		\$ 4,574,043	\$ 2,691,980	\$ 1,882,063	41%
2013-Feb		\$ 3,333,840	\$ 1,925,155	\$ 1,408,685	42%
2013-Mar		\$ 4,266,053	\$ 2,521,409	\$ 1,744,645	41%
2013-Apr		\$ 2,498,576	\$ 1,417,551	\$ 1,081,024	43%
2013-May		\$ 3,533,538	\$ 2,040,086	\$ 1,493,452	42%
2013-Jun		\$ 4,115,434	\$ 2,386,136	\$ 1,729,298	42%
2013-Jul		\$ 2,696,222	\$ 1,515,881	\$ 1,180,341	44%
2013-Aug		\$ 3,792,982	\$ 2,165,853	\$ 1,627,129	43%
2013-Sep		\$ 4,087,106	\$ 2,395,942	\$ 1,691,164	41%
2013-Oct		\$ 2,917,027	\$ 1,699,705	\$ 1,217,322	42%
2013-Nov		\$ 3,647,346	\$ 2,161,120	\$ 1,486,225	41%
2013-Dec		\$ 3,291,823	\$ 1,925,886	\$ 1,365,936	41%
2014-Jan		\$ 4,114,861	\$ 2,363,597	\$ 1,751,264	43%
2014-Feb		\$ 3,198,718	\$ 1,732,256	\$ 1,466,461	46%
2014-Mar		\$ 3,789,271	\$ 2,131,698	\$ 1,657,573	44%
2014-Apr		\$ 3,575,329	\$ 2,035,458	\$ 1,539,871	43%
2014-May		\$ 3,541,237	\$ 2,015,104	\$ 1,526,133	43%
2014-Jun		\$ 3,705,966	\$ 2,063,802	\$ 1,642,164	44%



作为最佳实践，请确保您的数据格式正确。在该情况下，在每个列中我们将更改**标签**以表示计算。在具有货币值的列中，我们将**数字格式**更改为**货币**，并将**格式样式**更改为 \$ #,##0;- \$ #,##0。将例如百分比的**数字格式**设置为**数字**，并将**格式设定**设置为**简单**和 **12%**。

您可以看到基于销售额和成本的每个月的计算的利润值。您还可以看到销售额有多少百分比构成利润值。

在应用程序数据中，我们已经有每月利润值的数据。这是在初始数据和我们的计算结果之间进行比较的不错机会。

执行以下操作：

1. 单击**添加列**并选择**度量**。
2. 单击 ***fx*** 符号。  
表达式编辑器打开。
3. 输入以下内容：Sum(Margin)

- 用表达式另外添加一个度量： $(Sum(Sales) - Sum(Cost)) - Sum(Margin)$
- 单击应用。

利润值表格具有来自数据集的每月利润值的额外列，及其与计算的利润值的差值。

Margin						
Month	Q	Sum(Sales)	Sum(Cost)	Calculated Margin	Margin %	Sum(Margin)
Totals		\$ 104,852,675	\$ 61,571,565	\$ 43,281,110	41%	\$ 43,253,189
2012-Jan		\$ 1,773,750	\$ 1,122,474	\$ 651,276	37%	\$ 651,276
2012-Feb		\$ 3,867,568	\$ 2,352,955	\$ 1,514,613	39%	\$ 1,514,613
2012-Mar		\$ 3,892,195	\$ 2,339,154	\$ 1,553,041	40%	\$ 1,553,041
2012-Apr		\$ 3,660,634	\$ 2,241,036	\$ 1,419,598	39%	\$ 1,419,598
2012-May		\$ 3,191,648	\$ 1,961,629	\$ 1,230,019	39%	\$ 1,230,019
2012-Jun		\$ 4,259,260	\$ 2,540,976	\$ 1,718,284	40%	\$ 1,718,284
2012-Jul		\$ 2,519,873	\$ 1,488,274	\$ 1,031,598	41%	\$ 1,031,598
2012-Aug		\$ 3,799,274	\$ 2,312,303	\$ 1,486,971	39%	\$ 1,486,971
2012-Sep		\$ 3,739,098	\$ 2,239,469	\$ 1,499,629	40%	\$ 1,499,629
2012-Oct		\$ 3,036,456	\$ 1,897,354	\$ 1,139,102	38%	\$ 1,139,102
2012-Nov		\$ 3,528,099	\$ 2,193,961	\$ 1,334,138	38%	\$ 1,334,138
2012-Dec		\$ 2,905,449	\$ 1,693,359	\$ 1,212,089	42%	\$ 1,212,089
2013-Jan		\$ 4,574,043	\$ 2,691,980	\$ 1,882,063	41%	\$ 1,882,063
2013-Feb		\$ 3,333,840	\$ 1,925,155	\$ 1,408,685	42%	\$ 1,408,685
2013-Mar		\$ 4,266,053	\$ 2,521,409	\$ 1,744,645	41%	\$ 1,744,645
2013-Apr		\$ 2,498,576	\$ 1,417,551	\$ 1,081,024	43%	\$ 1,081,024
2013-May		\$ 3,533,538	\$ 2,040,086	\$ 1,493,452	42%	\$ 1,493,452
2013-Jun		\$ 4,115,434	\$ 2,386,136	\$ 1,729,298	42%	\$ 1,729,298
2013-Jul		\$ 2,696,222	\$ 1,515,881	\$ 1,180,341	44%	\$ 1,180,341
2013-Aug		\$ 3,792,982	\$ 2,165,853	\$ 1,627,129	43%	\$ 1,627,129
2013-Sep		\$ 4,087,106	\$ 2,395,942	\$ 1,691,164	41%	\$ 1,691,164
2013-Oct		\$ 2,917,027	\$ 1,699,705	\$ 1,217,322	42%	\$ 1,217,322
2013-Nov		\$ 3,647,346	\$ 2,161,120	\$ 1,486,225	41%	\$ 1,486,225
2013-Dec		\$ 3,291,823	\$ 1,925,886	\$ 1,365,936	41%	\$ 1,365,936
2014-Jan		\$ 4,114,861	\$ 2,363,597	\$ 1,751,264	43%	\$ 1,731,437
2014-Feb		\$ 3,198,718	\$ 1,732,256	\$ 1,466,461	46%	\$ 1,463,099
2014-Mar		\$ 3,789,271	\$ 2,131,698	\$ 1,657,573	44%	\$ 1,657,573
2014-Apr		\$ 3,575,329	\$ 2,035,458	\$ 1,539,871	43%	\$ 1,537,112
2014-May		\$ 3,541,237	\$ 2,015,104	\$ 1,526,133	43%	\$ 1,526,133
2014-Jun		\$ 3,705,966	\$ 2,063,802	\$ 1,642,164	44%	\$ 1,640,192

计算的利润值列中的一些值不同于直接源自我们数据的利润值列中的值。利润值差异栏清楚地表明，这是在 2014 年期间的月份中发生的。计算出的利润值和数据集得出的利润值之间的差异很小，但它发生在特定年份，这一事实产生了一些问题。在这个年份有什么发生了变化？调查数据并提出正确的问题可能对您的业务很重要。

## 7.2 开发票延迟

在本例中，我们将使用基于一家公司的数据，该数据收集发票的创建日期和所生产货物的承诺交货日期。这两个日期并不总是一样的。另外一些发票可能具有两个承诺的交货日期。最早日期始终与发票日期相同，由公司使用的开发票系统自动创建。最晚承诺交货日期是公司与客户约定交货的日期。

我们先把这些日期加在一张表上。

在 *Examples from real life* 工作表上您将找到标题为 *Invoicing delays* 的表格。

执行以下操作：

1. 选择标题为 Invoicing delays 的可用表格。  
属性面板打开。
2. 单击**添加列**并选择**度量**。
3. 单击 *fx* 符号。  
表达式编辑器打开。
4. 输入以下内容：*Only([Invoice Date])*
5. 用表达式另外添加一个度量：*Max([Promised Delivery Date])*
6. 单击**应用**。

示出承诺的交货日期和每个发票的发票日期的表格

Invoicing delays		
Invoice Number	Invoice date	Promised delivery date
<b>Totals</b>	-	<b>31 Dec 2014</b>
100001	30 Apr 2013	29 Apr 2013
100002	30 Apr 2013	30 Apr 2013
100005	30 Apr 2013	30 Apr 2013
100006	30 Apr 2013	30 Apr 2013
100007	30 Apr 2013	30 Apr 2013
100008	30 Apr 2013	30 Apr 2013
100009	30 Apr 2013	30 Apr 2013
100010	30 Apr 2013	30 Apr 2013
100011	01 May 2013	01 May 2013
100013	01 May 2013	01 May 2013
100018	02 May 2013	02 May 2013
100021	02 May 2013	02 May 2013
100023	02 May 2013	02 May 2013
100027	03 May 2013	03 May 2013
100028	03 May 2013	03 May 2013
100029	03 May 2013	03 May 2013
100030	03 May 2013	03 May 2013
100034	06 May 2013	06 May 2013
100036	06 May 2013	06 May 2013



作为最佳实践，请确保您的数据格式正确。在显示日期的列中，将**数字格式**设置为**日期**，并将**格式**设置为**简单**和**2014 年 2 月 17 日**。

您可以看到发票日期和承诺的交货日期并非始终相同。当有两个承诺的交货日期时，我们需要使用最晚的一个来计算。

让我们计算一下发票日期和承诺交货日期之间的差值。我们将使用以下表达式：

**Max** ([Promised Delivery Date]) - [Invoice Date]

有三个场景：

- 两个日期相同，并且表达式的结果为 0。
- 承诺产品在创建发票之后交货，并且结果是正整数。

- 在承诺产品交货之后创建发票，并且结果是负整数。

执行以下操作：

- 单击**添加列**并选择**度量**。
- 单击 ***fx*** 符号。  
表达式编辑器打开。
- 输入以下内容：`Max([Promised Delivery Date])-[Invoice Date]`
- 单击**应用**。

表格显示每张发票的承诺交货日期和发票日期以及从开发票到承诺交货的天数

Invoicing delays			
Invoice Number	Invoice date	Promised delivery date	Days from invoicing to delivery
<b>Totals</b>	-	<b>31 Dec 2014</b>	-
307258	21 Jul 2012	22 Feb 2012	-150
108707	30 Jul 2013	29 Apr 2013	-92
109851	09 Aug 2013	14 May 2013	-87
111190	26 Aug 2013	31 May 2013	-87
112112	05 Sep 2013	10 Jun 2013	-87
116817	28 Oct 2013	16 Aug 2013	-73
109998	12 Aug 2013	05 Jun 2013	-68
113609	23 Sep 2013	22 Jul 2013	-63
115559	14 Oct 2013	12 Aug 2013	-63
108081	22 Jul 2013	21 May 2013	-62
109357	05 Aug 2013	05 Jun 2013	-61
310525	26 Aug 2012	26 Jun 2012	-61
315709	25 Oct 2012	25 Aug 2012	-61
329238	27 Dec 2012	27 Oct 2012	-61
103809	03 Jun 2013	08 Apr 2013	-56
112368	09 Sep 2013	16 Jul 2013	-55
118091	11 Nov 2013	18 Sep 2013	-54
112120	05 Sep 2013	15 Jul 2013	-52
112121	05 Sep 2013	18 Jul 2013	-49



根据最后列将表格排序，命名为 *Days from invoicing to delivery*。

在日期之间存在一系列的差值。负值表示发票延迟。负值表示承诺的交付在创建发票后完成。

让我们计算在承诺的交货日期之后创建的发票数。

执行以下操作：

- 单击**添加列**并选择**度量**。
- 单击 ***fx*** 符号。  
表达式编辑器打开。
- 输入以下内容：`Count(Distinct If(Aggr(Max([Promised Delivery Date])<[Invoice Date],[Invoice Number]), [Invoice Number]))`
- 单击**应用**。



或者我们可使用  $\text{Sum}(\text{Aggr}(\text{If}(\text{Max}([\text{Promised Delivery Date}]) - [\text{Invoice Date}] < 0, 1, 0), [\text{Invoice Number}]))$ 。

开发票延迟表格以额外列出延迟的发票数。

Invoicing delays					
Invoice Number	Q	Invoice date	Promised delivery date	Days from invoicing to delivery	Invoice delayed (T/F)
<b>Totals</b>		-	<b>31 Dec 2014</b>	-	<b>3421</b>
307258		21 Jul 2012	22 Feb 2012	-150	1
108707		30 Jul 2013	29 Apr 2013	-92	1
109851		09 Aug 2013	14 May 2013	-87	1
111190		26 Aug 2013	31 May 2013	-87	1
112112		05 Sep 2013	10 Jun 2013	-87	1
116817		28 Oct 2013	16 Aug 2013	-73	1
109998		12 Aug 2013	05 Jun 2013	-68	1
113609		23 Sep 2013	22 Jul 2013	-63	1
115559		14 Oct 2013	12 Aug 2013	-63	1
108081		22 Jul 2013	21 May 2013	-62	1
109357		05 Aug 2013	05 Jun 2013	-61	1
310525		26 Aug 2012	26 Jun 2012	-61	1
315709		25 Oct 2012	25 Aug 2012	-61	1
329238		27 Dec 2012	27 Oct 2012	-61	1
103809		03 Jun 2013	08 Apr 2013	-56	1
112368		09 Sep 2013	16 Jul 2013	-55	1
118091		11 Nov 2013	18 Sep 2013	-54	1
112120		05 Sep 2013	15 Jul 2013	-52	1
112121		05 Sep 2013	18 Jul 2013	-49	1
113609		23 Sep 2013	22 Jul 2013	-63	1

最后列作为发票总数百分比形式的 KPI 将更有意义。

执行以下操作：

1. 创建 KPI。
2. 单击**添加度量项**。单击 **fx** 符号。  
表达式编辑器打开。
3. 输入以下内容： $\text{Count}(\text{Distinct If}(\text{Aggr}(\text{Max}([\text{Promised Delivery Date}]) < [\text{Invoice Date}], [\text{Invoice Number}]), [\text{Invoice Number}])) / \text{Count}([\text{Invoice Number}])$
4. 单击**应用**。

KPI 示出延迟发票的百分比。

Percentage of delayed invoices

4%

让我们计算开发票中的平均延迟。

执行以下操作：

1. 新建 KPI。
2. 单击**添加度量项**。单击 **fx** 符号。  
表达式编辑器打开。

3. 输入以下内容：`Avg(Aggr(If(Max([Promised Delivery Date])<[Invoice Date]),(Max([Promised Delivery Date])-[Invoice Date])), [Invoice Number]))`
4. 单击**应用**。

KPI 示出发票中的平均延迟



### 7.3 谢谢！

---

现在，您已完成本教程的学习，希望您已获得有关 Qlik Sense 中图表表达式的一些基本知识。请访问我们的网站获取更多应用程序的灵感。