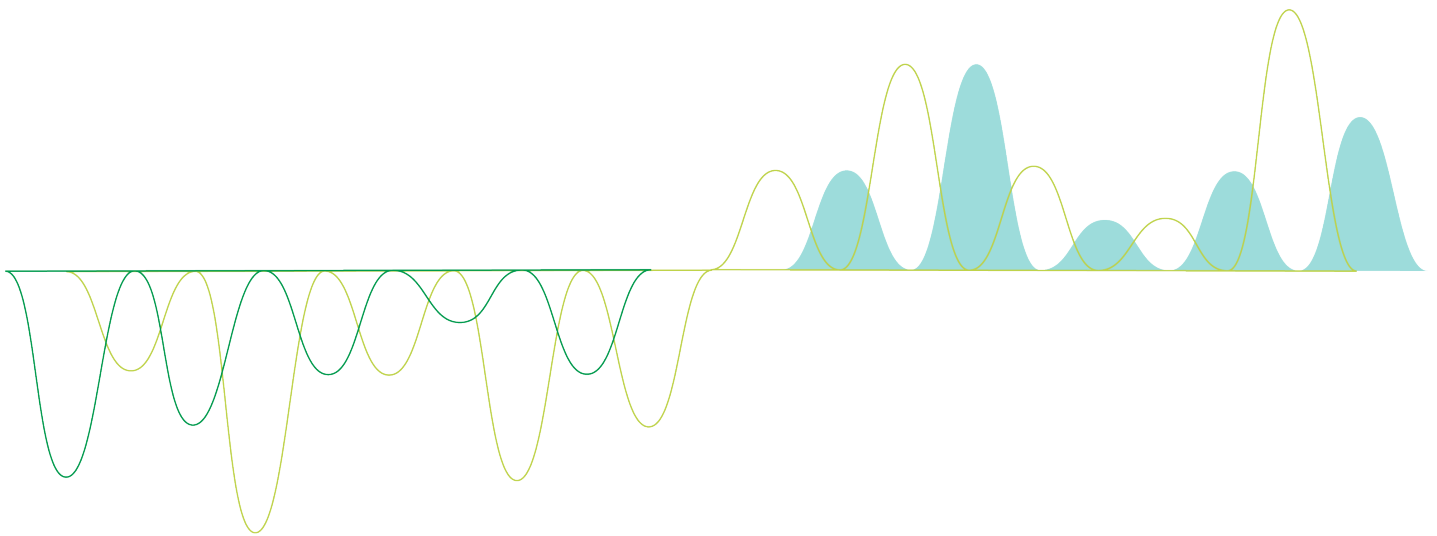


チュートリアル - チャートの数式

Qlik Sense®

May 2023

Copyright © 1993-2023 QlikTech International AB. All rights reserved.



1 チュートリアルへようこそ!	4
1.1 学習内容	4
1.2 このチュートリアルを完了する必要がある対象者	4
1.3 このチュートリアルでのレッスン	4
1.4 追加の資料とリソース	4
2 ビジュアライゼーションにおける数式の使用	5
2.1 数式とは?	5
2.2 数式をどこで使用できるか?	5
2.3 数式はいつ評価されるか	5
3 どの集計関数を使用するかについて	6
3.1 Sum() を使用した金額の統合	6
3.2 Max() を使用した最高売上高の計算	7
3.3 Min() を使用した最低売上高の計算	8
3.4 Count() を使用してエンティティの数をカウント	8
Count() と Count(distinct) の差分	9
4 ネストされた集計関数	11
4.1 関数では常に1つのレベルの集計	11
4.2 ネストされた集計に Aggr() を使用	11
4.3 最大平均注文値の計算	12
5 ネイキッド項目参照	15
5.1 数式で常に集計関数を使用	15
If() 関数を使用した請求書日付の分割	15
5.2 ネイキッド項目参照の回避	16
If() 関数でのネイキッド項目参照の回避	16
6 The importance of Only()	19
6.1 Only() を使用したさまざまな数式	20
7 実例	25
7.1 グロス マージンの割合の計算	25
7.2 請求の遅延	27
7.3 お疲れ様でした!	31

1 チュートリアルへようこそ!

このチュートリアルでは、Qlik Sense 内のチャートの数式を紹介します。数式は、関数、項目、数学演算子の組み合わせであり、データを処理して、視覚化で確認できる結果を生成するために使用されます。

チャートの数式は、多くの場合 メジャーで使用します。タイトルやサブタイトル、脚注、さらには軸などの数式を使用することで、より動的で効果的なビジュアライゼーションを作成することもできます。

1.1 学習内容

チュートリアルを完了した後は、ビジュアライゼーションで問題な数式を使用できるようになるでしょう。

1.2 このチュートリアルを完了する必要がある対象者

Qlik Sense の基本を習得している必要があります。例えば、さまざまなシートへのデータのロード、アプリの作成、ビジュアライゼーションの作成などを経験している必要があります。

データロードエディターへのアクセス権があり、Qlik Sense Enterprise on Windows におけるデータのロードが許可されている必要があります。

1.3 このチュートリアルでのレッスン

このチュートリアル内のトピックは、任意の順序で完了できます。ただし、後半のトピックは、その前のトピックに精通していることを前提としています。スクリーンショットは、Qlik Sense Enterprise SaaS で撮影したものです。別の展開で Qlik Sense Enterprise を使用している場合は、視覚的に多少の相違が生じることがあります。

1.4 追加の資料とリソース

- [Qlik](#) では、さらなる詳細情報を提供する、広範なリソースをご用意しています。
- [Qlik オンラインヘルプ](#) を利用できます。
- 無料のオンライン コースを含むトレーニングは、[Qlik Continuous Classroom](#) で利用できます。
- ディスカッション フォーラム、ブログなどは、[Qlik Community](#) にあります。

2 ビジュアライゼーションにおける数式の使用

Qlik Sense のビジュアライゼーションは、軸とメジャーで構築されたチャートから構築されています。関数により、ビジュアライゼーションを動的かつ複合的にすることができます。

情報をわかりやすく伝えるため、ビジュアライゼーションにはタイトルやサブタイトル、脚注、その他の要素を追加できます。ビジュアライゼーションを構成する要素は単純でも構いません。例：データを表す項目からなる軸、そしてテキストからなるタイトル。

メジャーは、項目に基づいて計算されます。例：**Sum(Cost)** は、項目 **Cost** の値すべてが、関数 **Sum** を使用して集計されていることを表します。つまり、**Sum(Cost)** 数式となります。

2.1 数式とは？

数式は関数および項目、数学演算子(+*/=)を組み合わせたものです。数式は、ビジュアライゼーションで確認可能な結果を生成するために、アプリでデータを処理する際に使用します。数式には、基本的な計算で構成されるシンプルなものから、関数項目と演算子を含む複雑なものまで、さまざまなタイプがあります。数式は、スクリプトとチャートのビジュアライゼーションで使用されます。

すべてのメジャーは、数式です。メジャーと異なり、数式には名前も説明データも存在しません。

軸、タイトル、サブタイトル、脚注などの数式を使用することで、より動的で効果的なビジュアライゼーションを作成できます。例えば、ビジュアライゼーションのタイトルを静的テキストではなく、選択内容によって結果が変わる数式からビジュアライゼーションのタイトルを生成できることを意味します。

2.2 数式をどこで使用できるか？

ビジュアライゼーションを編集する際、プロパティパネルに **fx** シンボルがある場合は、数式を使用できます。をクリックすると **fx** 数式エディタが開き、数式の作成と編集ができます。数式は、数式項目に直接入力することも可能です。

数式は、マスターアイテムとして直接保存することはできません。しかし、マスターメジャーとマスター軸に数式を含めることはできます。数式が使用されているメジャーまたは軸をマスターアイテムとして保存すると、メジャーまたは軸にある数式を保持できます。

2.3 数式はいつ評価されるか

ロードスクリプトでは、スクリプトの実行によって数式が評価されます。ビジュアライゼーションでは、数式に含まれる項目、変数、関数のいずれかで値または論理ステータスが変更されると、自動的に数式が評価されます。スクリプト式とチャート式では、構文および有効な関数にいくつか違いがあります。

3 どの集計関数を使用するかについて

集計関数は、多対1の関数です。多くのレコードの値を入力として使用し、これらをすべてのレコードを要約する単一の値に集約します。Sum(), Count(), Avg(), Min(), と Only() はすべて、集計関数です。

Qlik Sense では、方程式の大半で正確に1レベルの集計関数が必要です。そうした方程式には、チャートの数式、テキストボックス、ラベルが含まれます。数式に集計関数を含めないと、Qlik Sense が自動的に Only() 関数を割り当てます。

- 集計関数は、データ内の複数のレコードのいくつかのプロパティを説明する単一の値を返す関数です。
- 計算された軸を除く数式すべては、集計として評価されます。
- 数式内のすべての項目参照は、集計関数でラップする必要があります。



数式エディタを使用し、Qlik Sense 内で数式を作成したり変更したりできます。

3.1 Sum() を使用した金額の統合

Sum() は、集計データ全体の数式や項目による値の合計を計算します。

各マネージャーの総売上と、すべてのマネージャーの総売上を計算してみましょう。

アプリ内の Which Aggregations? シートには、Sum(), Max(), Min() というタイトルのテーブルと Count() というタイトルのテーブルの2つのテーブルがあります。各テーブルを使用して、集計関数を作成します。

次の手順を実行します。

1. 使用可能な Sum(), Max(), Min() テーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. **fx** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。Sum(Sales)
5. [適用] をクリックします。

Manager ごとの総売上を表示するテーブル

Sum(), Max (), Min()	
Manager	Sum(Sales)
Totals	\$ 104,852,674.81
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

各マネージャーの売上と、すべてのマネージャーの総売上を確認できます。



ベストプラクティスとして、データが適正に形式設定されていることを確認してください。この場合、**[数値書式]** は **[通貨]** に、**[書式パターン]** は **[\$ #,##0;-\$ #,##0]** に設定します。

3.2 Max() を使用した最高売上高の計算

Max() は集計データの行ごとの最高値を検出します。

次の手順を実行します。

1. **[列を追加]** をクリックして、**[メジャー]** を選択します。
2. ***fx*** 記号をクリックします。
数式エディタが開きます。
3. 以下を入力します。 **Max (Sales)**
4. **[適用]** をクリックします。

Manager ごとの総売上と最大売上を表示するテーブル

Sum(), Max (), Min()		
Manager	Sum(Sales)	Max(Sales)
Totals	\$ 104,852,674.81	\$ 555,376.00
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

各マネージャーの売上の最大値と合計の最大値がわかります。

3.3 Min() を使用した最低売上高の計算

Min() は集計データの行ごとの最低値を検出します。

次の手順を実行します。

1. [列を追加] をクリックして、[マネージャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。
3. 以下を入力します。Min (Sales)
4. [適用] をクリックします。

マネージャーごとの総売上、最大売上、最小売上を表示するテーブル

Sum(), Max (), Min()			
Manager	Sum(Sales)	Max(Sales)	Min(Sales)
Totals	\$ 104,852,674.81	\$ 555,376.00	-\$ 27,929.88
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

各マネージャーの売上の最小値と合計の最小値がわかります。

3.4 Count() を使用してエンティティの数をカウント

Count() は、各チャート軸に含まれる値、テキスト、数値の数のカウントに使われます。

データでは、各マネージャーが複数の営業担当者 ([営業担当者名]) の責任を負っています。営業担当者の数を計算してみましょう。

次の手順を実行します。

1. 使用可能な Count() テーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[マネージャー] を選択します。
3. **fx** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。Count([Sales Rep Name])
5. [適用] をクリックします。

営業担当者と営業担当者の総数を表示するテーブル。

Count()	
Sales Rep Name	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

営業担当者の総数は 64 であることがわかります。

Count() と Count(distinct) の差分

マネージャーの数を計算してみましょう。

次の手順を実行します。

1. テーブルに新しい軸を追加します。[マネージャー]。
単一のマネージャーが複数の営業担当者进行处理しているため、同じマネージャー名がテーブル内で複数回表示されます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. **fx** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。Count(Manager)
5. 次の数式を使用して別のメジャーを追加します。Count(distinct Manager)
6. [適用] をクリックします。

営業担当者、営業担当者の総数、各営業担当者の担当マネージャー、マネージャーの誤った総数、マネージャーの正しい総数を表示するテーブル。

3 どの集計関数を使用するかについて

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

数式として *Count(Manager)* を使用する列のマネージャーの総数が **64** と計算されたことがわかります。これは正しくありません。マネージャの総数は、*Count(distinct Manager)* 数式を使用して **18** として正しく計算されます。名前がリストにいくつあるかに関係なく、各マネージャーは 1 回だけカウントされます。

4 ネストされた集計関数

チャートの数式で利用される任意の項目名は、必ず1つの集計関数で囲む必要があります。集約をネストする必要がある場合は、**Aggr()** を使用して2番目の集約レベルを追加できます。**Aggr()** には、引数として集計関数が含まれています。

4.1 関数では常に1つのレベルの集計

一般的なアプリには、次が含まれます。

- データ内に 100 万レコード
- ピボットテーブル内に 100 行
- ゲージ、またはテキストボックス内の単一の KPI

影響の度合いは異なるにもかかわらず、3つの数値すべてがデータすべてを表す場合があります。数値は異なる集計レベルです。

集計関数は多くのレコードの値を入力として使用し、すべてのレコードの要約として表示できる1つの単一の値に集約します。制限が1つあります。別の集計関数内で集計関数を使用することはできません。通常は、すべての項目参照を1つの集計関数でラップする必要があります。

次の数式が機能します。

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

次の数式はネストされた集計であるため機能しません。

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

これに対する解決策は、**Aggr()** 関数の形で提供されます。その名前に反して、それは集約関数ではありません。これは「多対多」の関数であり、数学のマトリックスのようなものです。NレコードのあるテーブルをMレコードのあるテーブルに変換します。これは、値の配列を返します。また、1つのメジャーと1つまたは複数の軸を持つ仮想ストレートテーブルとみなすこともできます。



複数のレベルでネストされたチャート集計を作成する場合は、計算軸で**Aggr()** 関数を使用してください。

4.2 ネストされた集計に **Aggr()** を使用

Aggr() は、指定された軸上で計算された数式の値の配列を返します。たとえば、顧客別、地域別 sales の最大値です。高度な集計では、**Aggr()** 関数は別の集計関数で囲まれ、**Aggr()** の結果の配列をネストされる集計への入力として使用します。

これを使用すると、**Aggr()** ステートメントは仮想テーブルを生成し、1つの数式が1つ以上の軸によってグループ化されます。この仮想テーブルの結果は、外部集計関数によってさらに集計できます。

4.3 最大平均注文値の計算

チャートの数式で単純な **Aggr()** ステートメントを使用してみましょう。

全体的な指標を地域レベルで表示するだけでなく、さらに2つの複雑な数式も表示する必要があります。

- 各地域内のマネージャーによる最大平均注文値。
- その最大平均注文値を担当するマネージャー。

各地域の平均注文値は、標準の数式 **Sum(Sales)/Count([Order Number])** を使用して計算できます。

アプリ内の *Nested Aggregations* シートには、*Aggr() function* というタイトルのテーブルがあります。

次の手順を実行します。

1. 使用可能な **Aggr()** function テーブルを選択します。
プロパティパネルが開きます。
2. **[列を追加]** をクリックして、**[メジャー]** を選択します。
3. ***fx*** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。 **Sum(Sales)/Count([Order Number])**
5. **[適用]** をクリックします。

地域ごとの平均注文値を表示するテーブル。

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



ベストプラクティスとして、データが適正に形式設定されていることを確認してください。この場合は各列で、計算を表すように**[ラベル]**を変更します。金額の値を含む列では、**数値書式**を**Money**に、**書式パターン**を**\$ #,##0;- \$ #,##0**に変更します。

私たちの目標は、各地域の最大平均注文値を取得することです。**Aggr()** を使用して、マネージャーごとに各地域の平均注文値を取得し、それらの最大値を表示することを Qlik Sense に通知する必要があります。各地域のマネージャーごとの平均注文値を取得するには、**Aggr()** ステートメントに次の軸を含める必要があります。

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

この数式により Qlik Sense は、次のような仮想テーブルを生成します。

マネージャーごとの、各地域の平均注文値を表示する **Aggr()** 関数の仮想テーブル。

Virtual table of Aggr() function

Region	Manager	Average order value
Totals		-
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

Qlik Sense がマネージャーごとに各地域の個々の平均注文値を計算するとき、これらの値の最大値を見つける必要があります。これを行うには、**Aggr()** 関数を **Max()** でラップします。

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

次の手順を実行します。

1. [列を追加] をクリックして、[メジャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。
3. 以下を入力します。Max(Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))
4. [適用] をクリックします。

マネージャーごとの、地域、各地域の平均注文値、各地域の最大平均注文値を表示するテーブル。

Aggr() function		
Region	Average order value	Largest average order value
Totals	\$ 1,087	\$ 12,338
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

地域レベルで、すべてのマネージャーの最大平均注文値を確認できます。これは、2つの複雑な数式の1つ目です。次の要件は、これらの大きな平均注文値を担当するマネージャーの名前を値自体の横に表示することです。

これを行うには、以前のように同じ **Aggr()** 関数を使用しますが、今回は **FirstSortedValue()** 関数を併用します。**FirstSortedValue()** 関数は、関数の2番目の部分で指定された特定の軸について、マネージャーを提供するように Qlik Sense に指示します。

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



数式には小さな、しかし非常に重要な部分が1 つあります。**Aggr()** 数式の前にマイナス記号があります。**FirstSortedValue()** 関数内で、データの配列のソート順を指定できます。この場合、マイナス記号は Qlik Sense に最大から最小へのソートを指示します。

次の手順を実行します。

1. [列を追加] をクリックして、[メジャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。
3. 以下を入力します。 `FirstSortedValue(Manager,-Aggr(Sum(Sales)/ Count([Order Number]), Manager, Region))`
4. [適用] をクリックします。

地域、各地域の平均注文値、各地域の最大平均注文値、その注文値を担当するマネージャーを表示するテーブル。

Aggr() function			
Region	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 ネイキッド項目参照

集計関数で囲まれていない項目は、ネイキッドとみなされます。

ネイキッド項目参照は配列であり、複数の値を含んでいる可能性があります。そうであれば、Qlik Sense はそれを NULL として評価しますが、どの値が必要なのかを判別できません。

5.1 数式で常に集計関数を使用

数式による評価が正しくない場合は、集計関数がない可能性が大了。

数式内の項目参照は、値の配列です。例:

2 つのテーブル。1 つは **Max(Invoice Date)** が単一の値であることを表示し、もう1 つは **Invoice Date** が値の配列であることを表示します。

Max(Invoice Date)	Invoice Date
Max([Invoice Date])	Invoice Date Q
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

集計関数の項目 [請求書日付] を囲んで、1 つの値に集約する必要があります。

数式で集計関数を使用しない場合、Qlik Sense はデフォルトで **Only()** 関数を使用します。項目参照が複数の値を返す場合、項目は Qlik Sense により、NULL として解釈されます。

If() 関数を使用した請求書日付の分割

If() 関数は、条件付き集計によく使用されます。関数内の条件が True または False のどちらかに評価されるかによって異なる値を返します。

アプリ内の *Naked field references* シートには、*Using If() on Invoice dates* というタイトルのテーブルがあります。

次の手順を実行します。

1. Using If() on Invoice dates というタイトルの使用可能なテーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. **fx** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。If([Invoice Date]>= Date(41323), 'After', 'Before')
5. [適用] をクリックします。

参照日で分割された請求書日付を表示するテーブル。

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

この数式は、*Invoice Date* が参照日 2013 年 2 月 18 日より前かどうかをテストし、そうである場合は「Before」を返します。日付が参照日 2013 年 2 月 18 日以降の場合、「After」が返されます。参照日は整数 41323 で表されます。

5.2 ネイキッド項目参照の回避

一見すると、この数式は正しいように見えます。

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

参照日のあとに請求書日付を評価し、「After」を返すか、そうでなければ「Before」を返すはずですが、しかし、*Invoice Date* はネイキッド項目参照であり、集計関数を持たないため、いくつかの値を持つ配列であり、NULL として評価されます。前の例では、テーブル内の *Date* 値ごとに *Invoice Date* が 1 つしかなかったため、数式は正しく計算されました。

異なる軸の値の下で同様の数式が計算される方法と、ネイキッド項目参照の問題を解決する方法を見てみましょう。

If() 関数でのネイキッド項目参照の回避

以前と同様の数式を使用します。

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

今回は、参照日以降の売上を関数が合計します。

アプリ内の *Naked field references* シートには、*Sum(Amount)* というタイトルのテーブルがあります。

次の手順を実行します。

1. 使用可能な *Sum(Amount)* テーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. *fx* 記号をクリックします。
数式エディタが開きます。

- 以下を入力します。 `If([Invoice Date]>= 41323, Sum(Sales))`
- [適用] をクリックします。

年、各年の売上合計、**If()** 関数を使用した数式の結果を表示するテーブル。

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	-



[メジャー] の [ラベル] はそのままにして、各数式の違いを表示します。金額の値を含む列では、[数値書式] を [通貨] に、[書式パターン] を [\$ #,##0;-\$ #,##0] に変更します。

各年に、参照日より後の請求書日付の配列があります。数式には集計関数がないため、NULL と評価されます。正しい数式では、**If()** 関数の最初のパラメーターで **Min()** や **Max()** などの集計関数を使用する必要があります。

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

次の手順を実行します。

- [列を追加] をクリックして、[メジャー] を選択します。
- fx** 記号をクリックします。
数式エディタが開きます。
- 以下を入力します。 `If([Invoice Date]>= Date(41323), Sum(Sales))`
- [適用] をクリックします。

年、各年の売上合計、**If()** 関数を使用したさまざまな数式の結果を表示するテーブル。

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

または、**If()** 関数を **Sum()** 関数内に置くこともできます。

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

次の手順を実行します。

- [列を追加] をクリックして、[メジャー] を選択します。
- fx** 記号をクリックします。
数式エディタが開きます。

3. 以下を入力します。 $\text{Sum}(\text{If}([\text{Invoice Date}] \geq \text{Date}(41323), \text{Sales}))$
4. [適用] をクリックします。

年、各年の売上合計、**If()** 関数を使用したさまざまな式の結果を表示するテーブル。

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

最後から2番目の数式で、**If()** 関数は、軸の値ごとに1度評価されました。最後の数式では、生データの行ごとに1回評価されます。関数の評価方法の違いにより結果は異なりますが、どちらも答えを返します。最初の数式は単に NULL と評価されます。上の画像は、2013年2月18日を参照日として、数式の違いを表示しています。

6 The importance of Only()

グループ内に可能な値が1つしかない場合、**Only()** は値を返します。この値は、集計の結果になります。集計関数が指定さ入れていない場合、Qlik Sense は **Only()** が既定になります。

チャートの軸とパラメーターの間に1対1の関係がある場合、**Only()** 関数は返される可能性のある唯一の値を返します。複数の値がある場合は、NULL を返します。たとえば、単価 = 12 の製品を検索した場合、単価が12の製品が複数あると、NULL が返されます。

次の画像は、1対1と1対多の関係の相違を表しています。

Manager Number と *Manager* の間の1対1の関係を表示するテーブル

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

Sales Rep Name と *Manager* の1対多の関係を表示するテーブル。

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() 関数は、集計関数です。**Sum()** や **Count()** のように、この関数は多数のレコードを入力として使用して 1 つの値のみを返します。Qlik Sense は、ほぼすべての計算で集計を使用します。チャート、ソート数式、テキストボックス、高度な検索、計算されたラベルの数式はすべて集計であり、集計関数を使用しないと計算できません。

しかし、明示的な集計関数がない数式をユーザーが入力した場合はどうするのでしょうか。例えば、ソート数式が *Date* に設定されている場合はどうでしょうか？ または、`= [Product Type] = 'Beer and Wine'` という数式を使用してビールとワインの製品を購入した顧客の高度な検索がある場合はどうでしょうか？

このような場合、計算には、**Only()** 関数による影響が生じます。数式に明示的な集計関数がない場合、Qlik Sense は **Only()** 関数を暗黙的に使用します。上記の場合、ソート数式として **Only (Date)** が使用され、検索条件として **Only ([Product Type])='Beer and Wine'** が使用されます。

新しい数式が、ユーザーが予期しない結果を返す場合があります。上記の例はどちらも、*Date* または *Product Type* の値が 1 つしかない場合は機能しますが、値が複数ある場合はどちらも機能しません。

6.1 Only() を使用したさまざまな数式

類似の関数で KPI を 4 つ作成します。これにより、関数内にネイキッド項目参照がある場合、または異なる位置に **Only()** がある場合、選択の結果に与える影響の大きさを比較できます。

アプリ内の *Importance of Only()* シートには、軸が *Invoice Date* のフィルターペインがあります。

次の手順を実行します。

1. KPI を作成します。
2. [メジャーを追加] をクリックします。**fx** 記号 をクリックします。
数式エディタが開きます。
3. 以下を入力します。Month([Invoice Date])
4. さらに 3 つのメジャー付きの KPIs (Month(Only([Invoice Date])), Month(Max([Invoice Date])), Only(Month([Invoice Date]))) を作成します。
5. [適用] をクリックします。

4 つの KPIs と、4 つの異なるが類似した数式を表示するフィルター パネル。

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>	



各 KPI で、[数値の形式] は [数式の評価] に設定されています。

ネイキッド項目参照がある場合、**Only()** 関数は最下位レベルに挿入されます。つまり、最初の 2 つの KPI、Month([Invoice Date])、Month(Only([Invoice Date])) は同じように解釈され、常に同じ結果になります。

ご覧のとおり、KPI 4 つのうち 3 つが、NULL を返します。3 番目の KPI、Month(Max([Invoice Date])) は選択されていなくても、値を返しています。

数式を作成するときは、使用する集計、または複数の値がある場合はどの値を使用するかを常に自問する必要があります。NULL を使用して複数の値を表したい場合は、関数をそのままにしておきます。数には、**Sum()**、**Avg()**、**Min()**、または **Max()** を代わりに使用するほうがよいでしょう。文字列には、**Only()**、または **MinString()** を使用できるでしょう。

次の手順を実行します。

1. シートの編集を終了します。
2. フィルター パネルで、1 月の日付を選択します。
3. ✓ をクリックして選択を確認します。

KPI の結果は、単一の選択が行われたときに変化します。

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

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

単一の選択が行われると、すべての KPI が正しい答えを返します。数式に `Month([Invoice Date])` の数式などのネイキッド項目参照が含まれている場合でも、一意の選択を行っているため、適切な値を返すことができます。

次の手順を実行します。

1. フィルター パネルで、1 月の追加の日付を選択します。
2. ✓ をクリックして選択を確認します。

1 月に両方の日付で 2 つの選択が行われると、KPI の結果が変化します。

Month([Invoice Date]) —	Month(Only([Invoice Date])) —														
Month(Max([Invoice Date])) Jan	Only(Month([Invoice Date])) Jan														
<div> <input type="text"/> Invoice Date </div> <table> <tr><td>1/12/2012</td><td>✓</td></tr> <tr><td>1/13/2012</td><td>✓</td></tr> <tr><td>1/18/2012</td><td></td></tr> <tr><td>1/19/2012</td><td></td></tr> <tr><td>1/20/2012</td><td></td></tr> <tr><td>1/21/2012</td><td></td></tr> <tr><td>1/22/2012</td><td></td></tr> </table>		1/12/2012	✓	1/13/2012	✓	1/18/2012		1/19/2012		1/20/2012		1/21/2012		1/22/2012	
1/12/2012	✓														
1/13/2012	✓														
1/18/2012															
1/19/2012															
1/20/2012															
1/21/2012															
1/22/2012															

最初の2つのKPIは、NULLを返し、その他の2つのKPIは、1月の適切な日付を返します。具体的には、4番目のKPIは正しい答えを返します。これは、どちらの日付選択も1月の日付に対するものだからです。

次の手順を実行します。

1. フィルター パネルで、1月以外の月の追加の日付を選択します。
2. ✓ をクリックして選択を確認します。

異なる月の日付で複数の選択を行うと、KPIの結果が変化します。

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

異なる月の日付を使用して複数の選択が行われた場合、3番目の KPI のみが値を返します。数式 `Month(Max([Invoice Date]))` に従って、行われた選択から最大の月の値を返します。`Only()` はネイキッド項目参照を持つ数式に自動的に挿入されるため、数式にとって最低レベルが常に適切であるとは限りません。`Only()` の配置は重要です。

7 実例

Qlik Sense でのビジュアライゼーションにより、データの洞察が得ることができます。チャートで数式を使用すると、特定の業務に適用できる結果が得られます。Qlik Sense のさまざまな関数により、オプションが手軽に利用できない場合でも、必要に合わせて数式をカスタマイズできるようになります。

7.1 グロス マージンの割合の計算

グロス マージンとは、売上高と売上のためにかかった費用の差と定義します。月ごとのマージンと、月ごとの売上高に対するマージンの割合を計算します。

マージンの割合を計算するには、次の数式を使用します。

$$(\text{Sum}(\text{Sales}) - \text{Sum}(\text{Cost})) / \text{Sum}(\text{Sales})$$

数式はさらに簡略化することも可能です

$$1 - \text{Sum}(\text{Cost}) / \text{Sum}(\text{Sales})$$

アプリ内の *Examples from real life* シートには、*Margin* というタイトルのテーブルがあります。

次の手順を実行します。

1. *Margin* というタイトルの使用可能なテーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. **fx** 記号をクリックします。
数式エディタが開きます。
4. 以下を入力します。 $\text{Sum}(\text{Sales})$
5. 数式を使用して、さらに $\text{Sum}(\text{Cost})$ 、 $\text{Sum}(\text{Sales}) - \text{Sum}(\text{Cost})$ 、 $1 - \text{Sum}(\text{Cost}) / \text{Sum}(\text{Sales})$ の 3 つのメジャーを追加します。
6. [適用] をクリックします。

テーブルには、売上高、月ごとの費用の合計、そして月ごとに計算したマージンが金額と割合によって表示されます。

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%



ベストプラクティスとして、データが適正に形式設定されていることを確認してください。この場合は各列で、計算を表すように[ラベル]を変更します。金額の値を含む列では、**数値書式**を**Money**に、**書式パターン**を\$ #,##0;- \$ #,##0に変更します。マージンの割合の[数値書式]を[数値]に、そして[書式]を[単純]と[12%]に設定します。

売上高と費用に基づいて、月ごとに計算されたマージンが表示されます。マージンが売上高に占める割合も見るができます。

アプリデータには、すでに月ごとのマージンのデータが含まれています。元のデータと今回の計算結果を比較する良い機会です。

次の手順を実行します。

1. [列を追加] をクリックして、[メジャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。

3. 以下を入力します。Sum(Margin)
4. 次の数式を使用して別のメジャーを追加します。(Sum(Sales) - Sum(Cost)) - Sum(Margin)
5. [適用] をクリックします。

データセットからの月ごとのマージン向けに追加の列が加えられたマージンテーブルと、計算されたマージンとの差分。

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

計算されたマージン列の値の中に、データのマージン列から直接取得した値と異なるものがあります。マージンの不一致の列は、それが2014年のある月に起因することをはっきりと示しています。計算されたマージンとデータセットからのマージンの差分はわずかですが、それが特定の年に起因することには疑問が生じます。その年にはどのような変化があったのでしょうか。データを分析して相応しい質問をすることは、事業にとって重要な場合があります。

7.2 請求の遅延

この例では、請求書の作成とそれらが生産する商品の約束された納品の両方のために日付を収集する会社に基づくデータを使用します。2つの日付は常に同じとは限りません。さらに請求書によっては、約束の納品日が2つ存在する場合もあります。最も早い日付は、会社が使用する請求システムにより自動的に作成されるため、常に請求書日付と同日です。約束された最大の納品日は、会社とクライアントの間で配達が行われることに同意した日付です。

これらの日付をテーブルに追加することから始めましょう。

Examples from real life シートには、*Invoicing delays* というタイトルのテーブルがあります。

次の手順を実行します。

1. *Invoicing delays* というタイトルの使用可能なテーブルを選択します。
プロパティパネルが開きます。
2. [列を追加] をクリックして、[メジャー] を選択します。
3. ***fx*** 記号 をクリックします。
数式エディタが開きます。
4. 以下を入力します。Only([Invoice Date])
5. 次の数式を使用して別のメジャーを追加します。Max([Promised Delivery Date])
6. [適用] をクリックします。

テーブルは、各請求書の約束の納品日と請求書日付を示しています

Invoicing delays		
Invoice Number 	Invoice date	Promised delivery date
Totals	-	31 Dec 2014
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



ベストプラクティスとして、データが適正に形式設定されていることを確認してください。日付を表示する列で、[数値書式] を[日付] に、そして[書式] を[単純] と[2014 年 2 月 17 日] に設定します。

請求書日付と約束の納品日は必ずしも同じとは限らないことがわかります。約束の納品日が2つある場合は、最大の納品日を使用して計算する必要があります。

請求書日付と約束の納品日の差分を計算してみましょう。次の数式を使用します。

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

3つのシナリオが考えられます。

- 両方の日付は同日であり、数式の結果は 0。
- 請求書が作成された後に製品の納品が約束され、結果は正の整数。
- 製品の納品が約束された後に請求書が作成され、結果は負の整数。

次の手順を実行します。

1. [列を追加] をクリックして、[メジャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。
3. 以下を入力します。 $\text{Max}([\text{Promised Delivery Date}]) - [\text{Invoice Date}]$
4. [適用] をクリックします。

テーブルには、各請求書の約束の納品日と請求書日付、さらに請求から約束の納品までの日数が表示されています。

Invoicing delays			
Invoice Number	Invoice date	Promised delivery date	Days from invoicing to delivery
Totals	-	31 Dec 2014	-
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



請求から納品までの日数と名付けられた最後の列に基づいてテーブルをソートします。

日付の間には差分の範囲があります。請求書が遅れたことを示す値の間を移動します。正の数値は、請求書が作成された後に約束の納品があったことを示します。

約束の納品日の後に作成された請求書の数を計算しましょう。

次の手順を実行します。

1. [列を追加] をクリックして、[メジャー] を選択します。
2. **fx** 記号をクリックします。
数式エディタが開きます。

- 以下を入力します。Count(Distinct If(Aggr(Max([Promised Delivery Date])<[Invoice Date],[Invoice Number]),[Invoice Number]))
- [適用] をクリックします。



代わりに、[Sum(Aggr(If(Max([Promised Delivery Date])-[Invoice Date]< 0, 1, 0), [Invoice Number]))] を使用することもできます。

追加の列がある請求の遅延テーブルには、遅れた請求書の数が表示されています。

Invoicing delays					
Invoice Number	Q	Invoice date	Promised delivery date	Days from invoicing to delivery	Invoice delayed (T/F)
Totals		-	31 Dec 2014	-	3421
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
117469		04 Nov 2013	16 Sep 2013	-40	1

KPI は請求書の合計数の割合として、最後の列がより有意義になります。

次の手順を実行します。

- KPI を作成します。
- [メジャーを追加] をクリックします。fx 記号 をクリックします。
数式エディタが開きます。
- 以下を入力します。Count(Distinct If(Aggr(Max([Promised Delivery Date])<[Invoice Date],[Invoice Number]),[Invoice Number]))/Count([Invoice Number])
- [適用] をクリックします。

KPI は、遅れた請求書の割合を示しています。

Percentage of delayed invoices

4%

請求の遅延の平均を計算しましょう。

次の手順を実行します。

1. 新しい KPI を作成します。
2. [メジャーを追加] をクリックします。**fx** 記号 をクリックします。
数式エディタが開きます。
3. 以下を入力します。Avg(Aggr(If(Max([Promised Delivery Date])<[Invoice Date]), (Max([Promised Delivery Date])-[Invoice Date])), [Invoice Number]))
4. [適用] をクリックします。

KPI は、請求の遅延の平均を示しています



7.3 お疲れ様でした!

以上でチュートリアルは終了です。ここでご紹介した基本情報を Qlik Sense でのチャートの数式にお役立てください。ウェブサイトでは、アプリ作成のヒントやアイデアを公開しています。ぜひご利用ください。