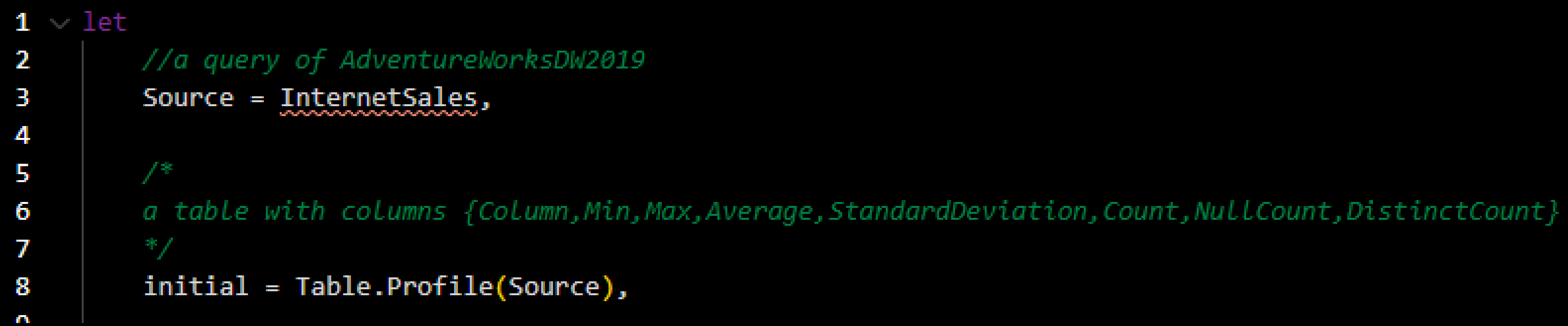
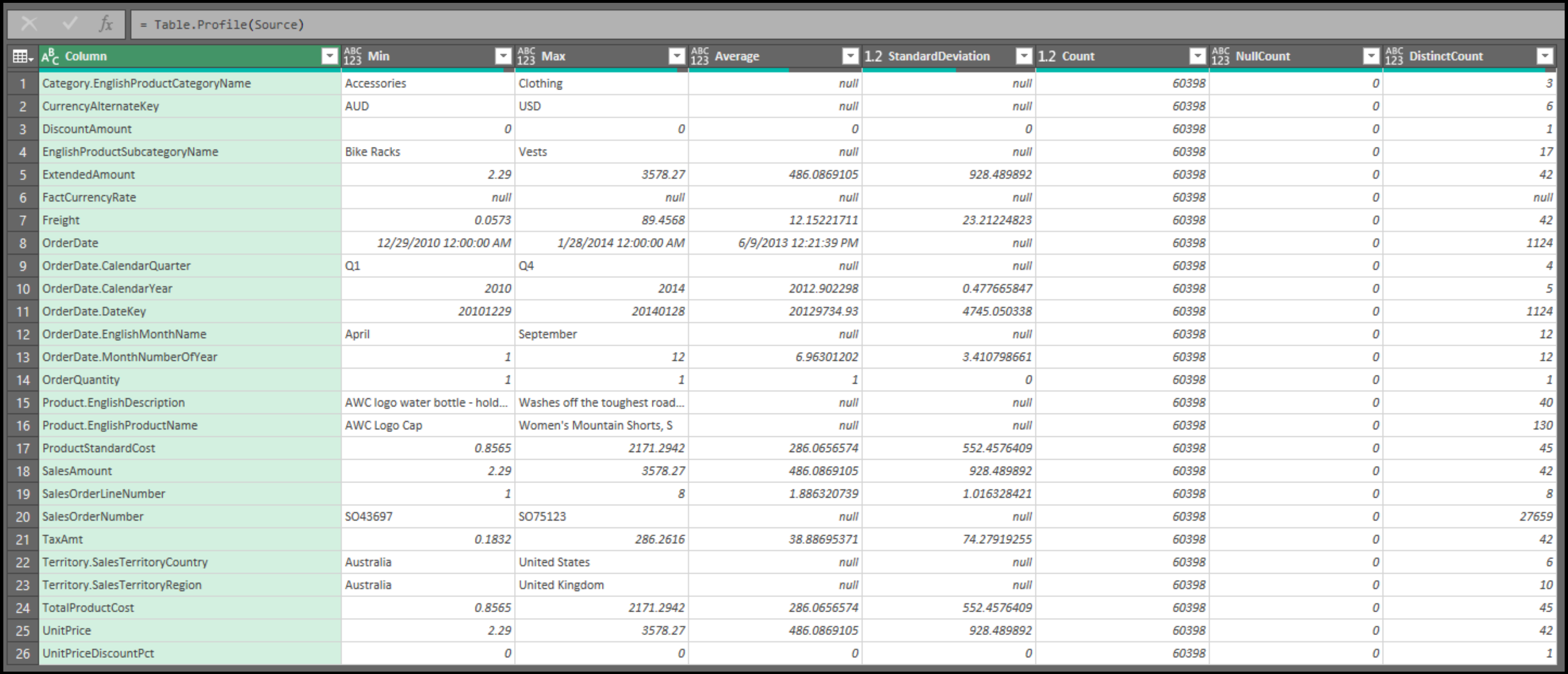
Power Query (M): Descriptive statistics using Table.Profile and custom functions

We can get some descriptive statistics of a table using Table.Profile



```
let
    //a query of AdventureWorksDW2019
    Source = InternetSales,

    /*
    a table with columns {Column,Min,Max,Average,StandardDeviation,Count,NullCount,DistinctCount}
    */
    initial = Table.Profile(Source),
```

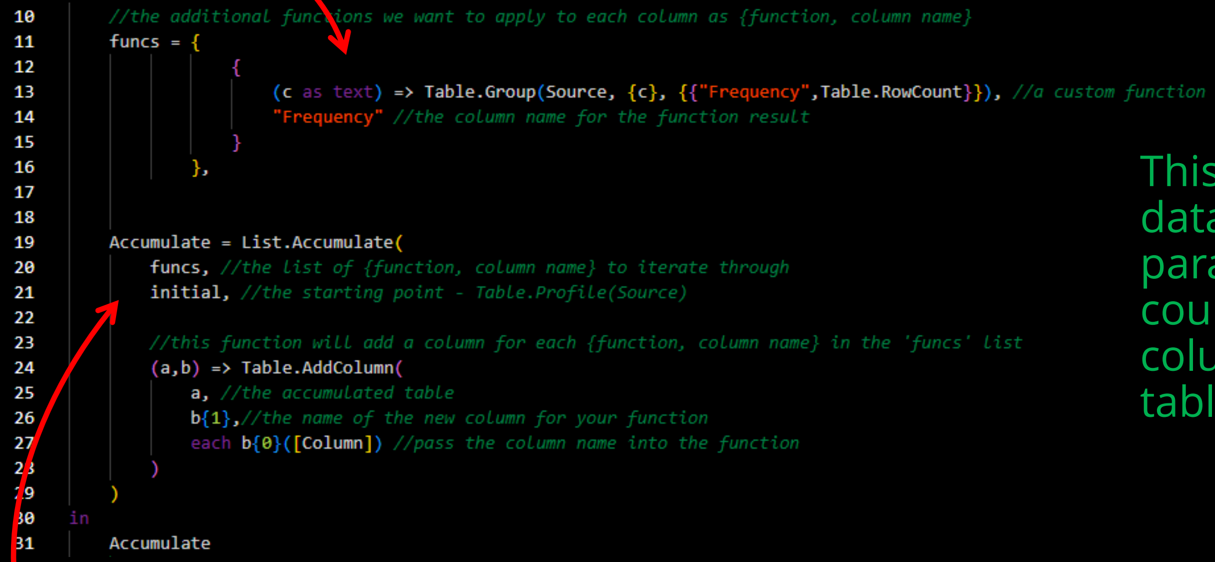


= Table.Profile(Source)

	Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount
1	Category.EnglishProductCategoryName	Accessories	Clothing	null	null	60398	0	3
2	CurrencyAlternateKey	AUD	USD	null	null	60398	0	6
3	DiscountAmount	0	0	0	0	60398	0	1
4	EnglishProductSubcategoryName	Bike Racks	Vests	null	null	60398	0	17
5	ExtendedAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42
6	FactCurrencyRate	null	null	null	null	60398	0	null
7	Freight	0.0573	89.4568	12.15221711	23.21224823	60398	0	42
8	OrderDate	12/29/2010 12:00:00 AM	1/28/2014 12:00:00 AM	6/9/2013 12:21:39 PM	null	60398	0	1124
9	OrderDate.CalendarQuarter	Q1	Q4	null	null	60398	0	4
10	OrderDate.CalendarYear	2010	2014	2012.902298	0.477665847	60398	0	5
11	OrderDate.DateKey	20101229	20140128	20129734.93	4745.050338	60398	0	1124
12	OrderDate.EnglishMonthName	April	September	null	null	60398	0	12
13	OrderDate.MonthNumberOfYear	1	12	6.96301202	3.410798661	60398	0	12
14	OrderQuantity	1	1	1	0	60398	0	1
15	Product.EnglishDescription	AWC logo water bottle - hold...	Washes off the toughest road...	null	null	60398	0	40
16	Product.EnglishProductName	AWC Logo Cap	Women's Mountain Shorts, S	null	null	60398	0	130
17	ProductStandardCost	0.8565	2171.2942	286.0656574	552.4576409	60398	0	45
18	SalesAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42
19	SalesOrderLineNumber	1	8	1.886320739	1.016328421	60398	0	8
20	SalesOrderNumber	SO43697	SO75123	null	null	60398	0	27659
21	TaxAmt	0.1832	286.2616	38.88695371	74.27919255	60398	0	42
22	Territory.SalesTerritoryCountry	Australia	United States	null	null	60398	0	6
23	Territory.SalesTerritoryRegion	Australia	United Kingdom	null	null	60398	0	10
24	TotalProductCost	0.8565	2171.2942	286.0656574	552.4576409	60398	0	45
25	UnitPrice	2.29	3578.27	486.0869105	928.489892	60398	0	42
26	UnitPriceDiscountPct	0	0	0	0	60398	0	1

If we want additional statistics, we can add them using List.Accumulate

A list of {function, result column name} pairs where the function accepts a column name from the source table and produces some result from it

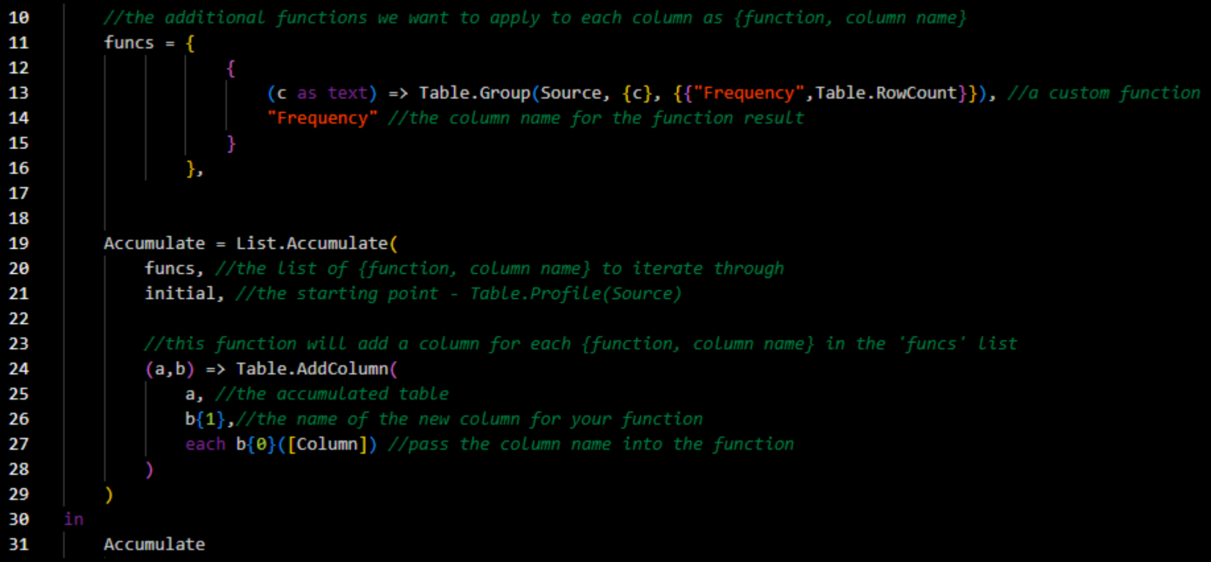


```
    //the additional functions we want to apply to each column as {function, column name}
    funcs = {
                {
                    (c as text) => Table.Group(Source, {c}, {{"Frequency",Table.RowCount}}), //a custom function
                    "Frequency" //the column name for the function result
                }
            },


    Accumulate = List.Accumulate(
        funcs, //the list of {function, column name} to iterate through
        initial, //the starting point - Table.Profile(Source)

        //this function will add a column for each {function, column name} in the 'funcs' list
        (a,b) => Table.AddColumn(
            a, //the accumulated table
            b{1},//the name of the new column for your function
            each b{0}([Column]) //pass the column name into the function
        )
    )
in
    Accumulate
```

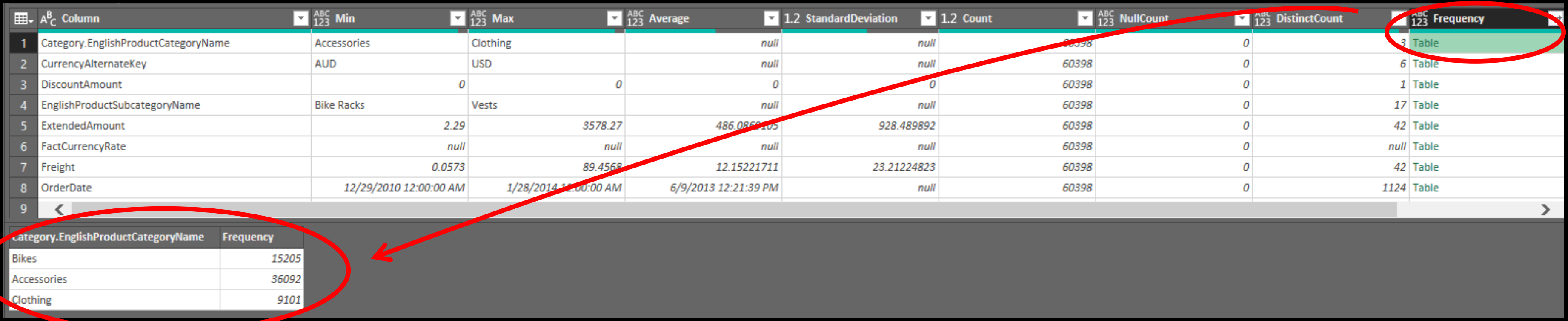
This function groups the Source data by column represented by the parameter c and calculates the row count of each unique item in that column, producing a frequency table for that column

Starting from the result of Table.Profile – initial -, iterate through the list of {function, result column name} and apply the function in the first element – b{0} - to each column in Source, whose name is in [Column] and give the new column the name in the second element – b{1}

We now have a frequency table for each column



```
    //the additional functions we want to apply to each column as {function, column name}
    funcs = {
                {
                    (c as text) => Table.Group(Source, {c}, {{"Frequency",Table.RowCount}}), //a custom function
                    "Frequency" //the column name for the function result
                }
            },


    Accumulate = List.Accumulate(
        funcs, //the list of {function, column name} to iterate through
        initial, //the starting point - Table.Profile(Source)

        //this function will add a column for each {function, column name} in the 'funcs' list
        (a,b) => Table.AddColumn(
            a, //the accumulated table
            b{1},//the name of the new column for your function
            each b{0}([Column]) //pass the column name into the function
        )
    )
in
    Accumulate
```



	Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount	Frequency
1	Category.EnglishProductCategoryName	Accessories	Clothing	null	null	60398	0	3	Table
2	CurrencyAlternateKey	AUD	USD	null	null	60398	0	6	Table
3	DiscountAmount	0	0	0	0	60398	0	1	Table
4	EnglishProductSubcategoryName	Bike Racks	Vests	null	null	60398	0	17	Table
5	ExtendedAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42	Table
6	FactCurrencyRate	null	null	null	null	60398	0	null	Table
7	Freight	0.0573	89.4568	12.15221711	23.21224823	60398	0	42	Table
8	OrderDate	12/29/2010 12:00:00 AM	1/28/2014 12:00:00 AM	6/9/2013 12:21:39 PM	null	60398	0	1124	Table
9									

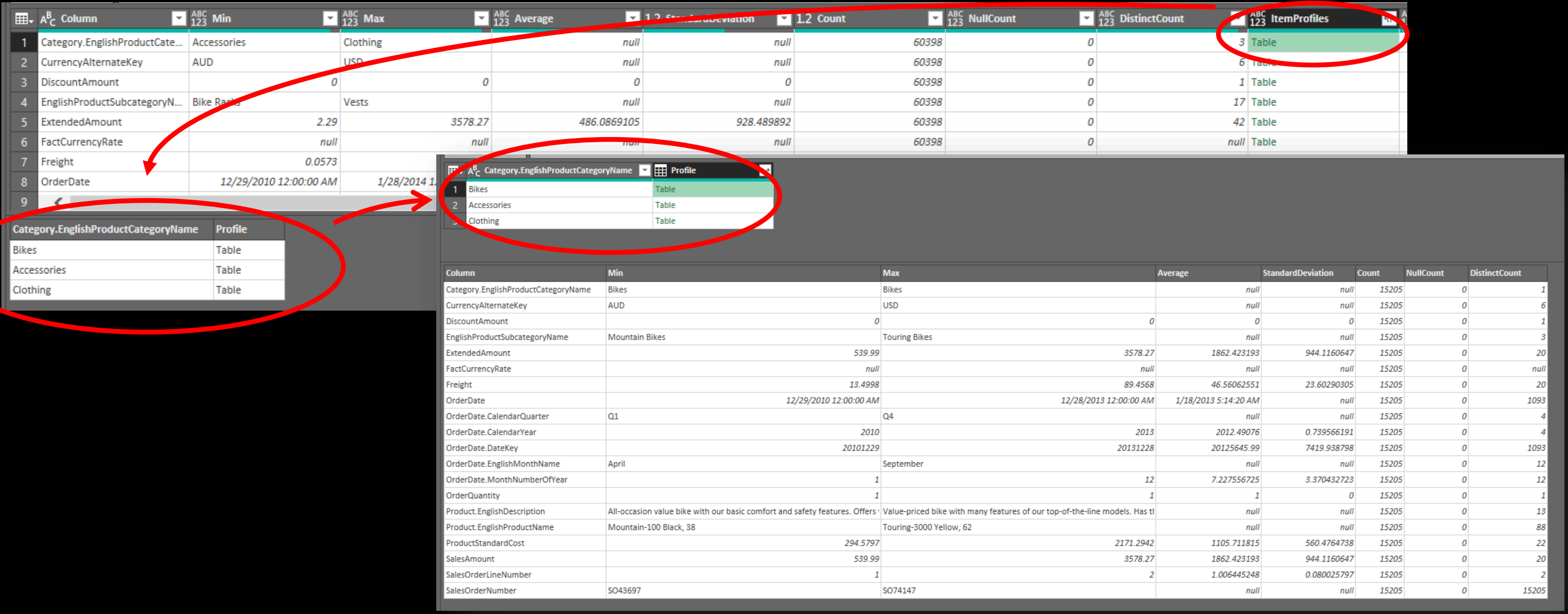
Category.EnglishProductCategoryName	Frequency
Bikes	15205
Accessories	36092
Clothing	9101

flexyourdata.com | youtube.com/@flexyourdata | linkedin.com/in/owenhprice

We can add as many functions as we want

Apply Table.Profile to each unique item in column with name in the parameter c

```
    funcs = {
            {(c as text) => Table.Group(Source,{c},{{"Frequency",Table.RowCount}}), "Frequency"},

            {(c as text) => Table.Group(Source,{c},{{"Profile",Table.Profile}})  , "ItemProfiles"}
        },
```



	Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount	ItemProfiles
1	Category.EnglishProductCate...	Accessories	Clothing	null	null	60398	0	3	Table
2	CurrencyAlternateKey	AUD	USD	null	null	60398	0	6	Table
3	DiscountAmount	0	0	0	0	60398	0	1	Table
4	EnglishProductSubcategoryN...	Bike Racks	Vests	null	null	60398	0	17	Table
5	ExtendedAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42	Table
6	FactCurrencyRate	null	null	null	null	60398	0	null	Table
7	Freight	0.0573							
8	OrderDate	12/29/2010 12:00:00 AM	1/28/2014 1						
9									

Category.EnglishProductCategoryName	Profile
Bikes	Table
Accessories	Table
Clothing	Table

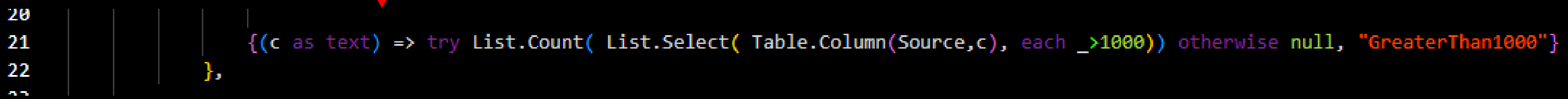
	Category.EnglishProductCategoryName	Profile
1	Bikes	Table
2	Accessories	Table
3	Clothing	Table

Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount
Category.EnglishProductCategoryName	Bikes	Bikes	null	null	15205	0	1
CurrencyAlternateKey	AUD	USD	null	null	15205	0	6
DiscountAmount	0	0	0	0	15205	0	1
EnglishProductSubcategoryName	Mountain Bikes	Touring Bikes	null	null	15205	0	3
ExtendedAmount	539.99	3578.27	1862.423193	944.1160647	15205	0	20
FactCurrencyRate	null	null	null	null	15205	0	null
Freight	13.4998	89.4568	46.56062551	23.60290305	15205	0	20
OrderDate	12/29/2010 12:00:00 AM	12/28/2013 12:00:00 AM	1/18/2013 5:14:20 AM	null	15205	0	1093
OrderDate.CalendarQuarter	Q1	Q4	null	null	15205	0	4
OrderDate.CalendarYear	2010	2013	2012.49076	0.739566191	15205	0	4
OrderDate.DateKey	20101229	20131228	20125645.99	7419.938798	15205	0	1093
OrderDate.EnglishMonthName	April	September	null	null	15205	0	12
OrderDate.MonthNumberOfYear	1	12	7.227556725	3.370432723	15205	0	12
OrderQuantity	1	1	1	0	15205	0	1
Product.EnglishDescription	All-occasion value bike with our basic comfort and safety features. Offers	Value-priced bike with many features of our top-of-the-line models. Has tl	null	null	15205	0	13
Product.EnglishProductName	Mountain-100 Black, 38	Touring-3000 Yellow, 62	null	null	15205	0	88
ProductStandardCost	294.5797	2171.2942	1105.711815	560.4764738	15205	0	22
SalesAmount	539.99	3578.27	1862.423193	944.1160647	15205	0	20
SalesOrderLineNumber	1	2	1.006445248	0.080025797	15205	0	2
SalesOrderNumber	SO43697	SO74147	null	null	15205	0	15205

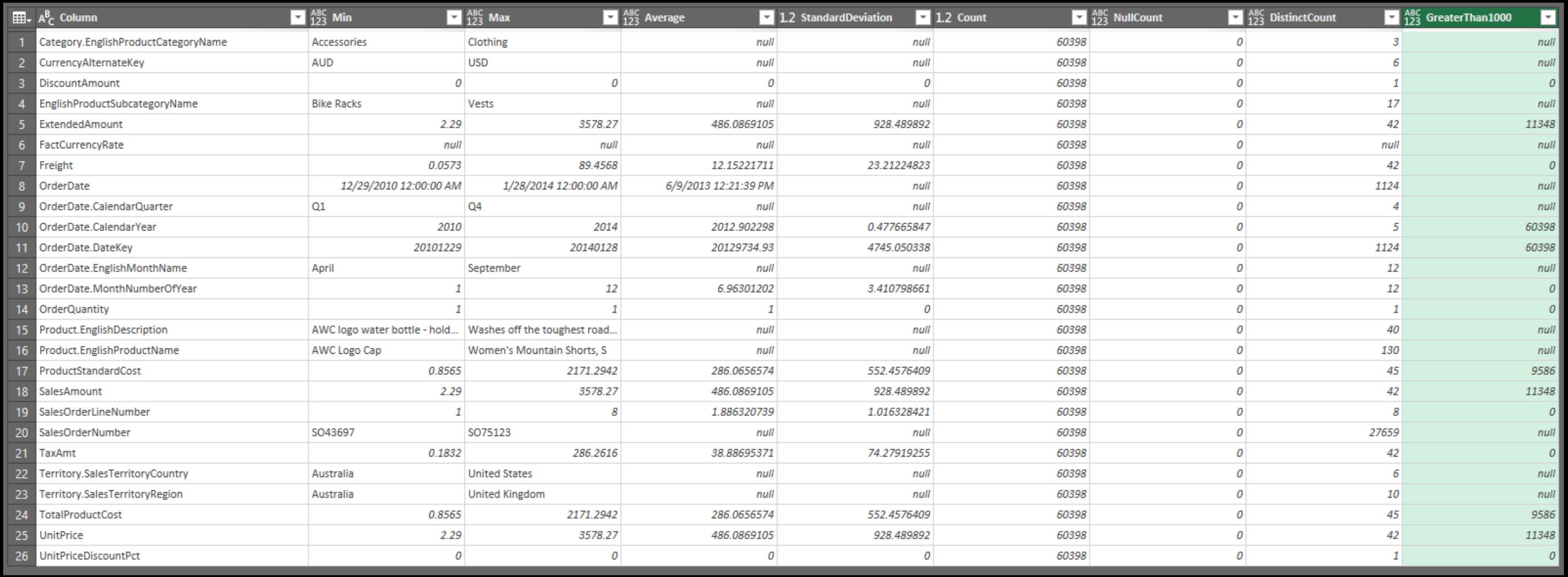
flexyourdata.com | youtube.com/@flexyourdata | linkedin.com/in/owenhprice

Defined however we want

This function counts rows in the column whose value is greater than 1000



```
                {(c as text) => try List.Count( List.Select( Table.Column(Source,c), each _>1000)) otherwise null, "GreaterThan1000"}
            },
```



	Column	Min	Max	Average	StandardDeviation	Count	NullCount	DistinctCount	GreaterThan1000
1	Category.EnglishProductCategoryName	Accessories	Clothing	null	null	60398	0	3	null
2	CurrencyAlternateKey	AUD	USD	null	null	60398	0	6	null
3	DiscountAmount	0	0	0	0	60398	0	1	0
4	EnglishProductSubcategoryName	Bike Racks	Vests	null	null	60398	0	17	null
5	ExtendedAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42	11348
6	FactCurrencyRate	null	null	null	null	60398	0	null	null
7	Freight	0.0573	89.4568	12.15221711	23.21224823	60398	0	42	0
8	OrderDate	12/29/2010 12:00:00 AM	1/28/2014 12:00:00 AM	6/9/2013 12:21:39 PM	null	60398	0	1124	null
9	OrderDate.CalendarQuarter	Q1	Q4	null	null	60398	0	4	null
10	OrderDate.CalendarYear	2010	2014	2012.902298	0.477665847	60398	0	5	60398
11	OrderDate.DateKey	20101229	20140128	20129734.93	4745.050338	60398	0	1124	60398
12	OrderDate.EnglishMonthName	April	September	null	null	60398	0	12	null
13	OrderDate.MonthNumberOfYear	1	12	6.96301202	3.410798661	60398	0	12	0
14	OrderQuantity	1	1	1	0	60398	0	1	0
15	Product.EnglishDescription	AWC logo water bottle - hold...	Washes off the toughest road...	null	null	60398	0	40	null
16	Product.EnglishProductName	AWC Logo Cap	Women's Mountain Shorts, S	null	null	60398	0	130	null
17	ProductStandardCost	0.8565	2171.2942	286.0656574	552.4576409	60398	0	45	9586
18	SalesAmount	2.29	3578.27	486.0869105	928.489892	60398	0	42	11348
19	SalesOrderLineNumber	1	8	1.886320739	1.016328421	60398	0	8	0
20	SalesOrderNumber	SO43697	SO75123	null	null	60398	0	27659	null
21	TaxAmt	0.1832	286.2616	38.88695371	74.27919255	60398	0	42	0
22	Territory.SalesTerritoryCountry	Australia	United States	null	null	60398	0	6	null
23	Territory.SalesTerritoryRegion	Australia	United Kingdom	null	null	60398	0	10	null
24	TotalProductCost	0.8565	2171.2942	286.0656574	552.4576409	60398	0	45	9586
25	UnitPrice	2.29	3578.27	486.0869105	928.489892	60398	0	42	11348
26	UnitPriceDiscountPct	0	0	0	0	60398	0	1	0

Takeaways

1. Table.Profile quickly creates basic descriptive statistics of every column in a table

2. We can use List.Accumulate and a list of {function, result column name} pairs to add custom profiling

3. We can also use Table.Profile in the custom function to profile the table grouped by the items within within each column