

# **DAX TWEAKS FOR GRAPHICAL DIVERSIFICATION OF THE REPORT**

**Power BI Next Step**

**Štěpán Rešl**

**2022**



(Steve)  
Štěpán Rešl



Data  
Brothers



JAK NA **POWER BI**



**Power BI**  
kafíčko



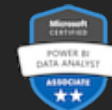
@tpnRel1



/štěpán-rešl-464084152



/data-god-stepan-resl



# Cheat Sheet

- Actual Pages
  - Overview
  - Power Query
  - DAX
- Where to find?
  - [www.kratosbi.com](http://www.kratosbi.com)

# OVERVIEW

## What is Power BI?

"It is Microsoft's Self-Service Business Intelligence tool for processing and analyzing data."

## Components

- Power BI Desktop**—Desktop application
  - Report**—Multi-page canvas visible to end users. It serves for the placement of visuals, buttons, images, slicers, etc.
  - Data**—Preview pane for data loaded into the model.
  - Model**—Editable scheme of relationships between tables in a model. Pages can be used in a model for easier navigation.
  - Power Query**—A tool for connecting, transforming, and combining data.
    - "Apart from the standard version, there is also a version for Report Server."
- Power BI Service**—A cloud service enabling access to, and sharing and administration of, output data.
  - Workspace**—There are three types of workspaces: **Personal**, **Team**, and **Developer** template app. They serve as storage and enable controlled access to output data.
  - Dashboard**—A space consisting of tiles in which visuals and report pages are stored.
  - Append**—A report of pages containing visuals.\*
  - Worksheet**—A published Excel worksheet. Can be used as a tile on a dashboard.
  - Dataset**—A published sequence for fetching and transforming data from Power BI Desktop.
  - Dataflow**—Online Power Query representing a special dataset outside of Power BI Desktop.\*
  - Application**—A single location combining one or more reports or dashboards.
  - Admin portal**—Administration portal that lets you configure capacities, permissions, and capabilities for individual users and workspaces.
    - Can be created and edited in the **Power BI Service gateway**.
  - Data Gateway**—On-premises data gateway that lets you transport data from an internal network or a custom device to the **Power BI Service**.
  - Power BI Mobile**—Mobile app for viewing reports. Mobile view is applied, if it exists, otherwise the desktop view is applied.
  - Report Server**—On-premises version of Power BI Service.
  - Report Builder**—A tool for creating page reports.

## Built-in and additional languages

**Built-in languages**

- M/Query language**—Lets you transform data in Power Query.
- DAX (Data Analysis Expressions)**—Lets you define custom calculated tables, columns, and measures in Power BI Desktop.
  - "Both languages are natively available in Power BI, which eliminates the need to install anything."

**Additional languages**

- Python**—Lets you fetch data and create visuals. Requires installation of the Python language on your computer and enabling Python scripting.
- R**—Lets you fetch and transform data and create visuals. Requires installation of the R language on your computer and enabling R scripting.

## Power Query

Works with data fetched from data sources using connectors. This data is then processed at the Power BI app level and stored to an in-memory database in the program background. This means that data is not processed at the source level. The basic unit in Power Query is **query**, which means one sequence consisting of steps. A step is a data command that dictates what should happen to the data when it is loaded into Power BI. The basic definition of each step is based on its use:

- Connecting data**—Each query begins with a function that provides data for the subsequent steps. E.g., data can be loaded from **Excel, SQL database, SharePoint** etc. Connection steps can also be used later.
- Transforming data**—Steps that modify the structure of the data. These steps include features such as **Pivot Column**, converting columns to rows, grouping data, splitting columns, removing columns, etc. Transformation steps are necessary in order to clean data from not entirely clean data sources.
- Combining data**—Data split into multiple source files needs to be combined so that it can be analyzed in bulk. Functions include merging queries and appending queries.
- Merge queries**—This function merges queries based on the selected key. The primary query then contains a column which can be used to extract data from a secondary query. Supports typical join types.
- Append query**—Places the resulting data from one or more selected queries under the primary query. In this case, data is placed in columns with names that are the same. Non-matching columns form new columns with a unique name in the primary query.
- Custom function**—A query intended to apply a pre-defined sequence of steps so that the author does not need to create them repeatedly. The custom function can also accept input data (values, sheets, etc.) to be used in the sequence.
- Parameter**—Values independent of datasets. These values can then be used in queries. Values enable the quick editing of a model because they can be changed in the Power BI Service environment.

## Dataflow

The basic unit is a table or Entity consisting of columns or **Fields**. Just like Queries in Power Query, Entities in Dataflows consist of sequences of steps. The result of such steps is stored in native Azure Data Lake Gen 2.

"You can connect a custom Data Lake where the data will be stored."

There are three types of entities:

- Standard entity**—It only works with data fetched directly from a data source and with data from non-stored entities within the same dataflow.
- Computed entity**—It uses data from another stored entity within the same dataflow.
- Linked entity**—Uses data from an entity located in another dataflow. If data in the original entity is updated, the new data is directly passed to all linked entities.

"Can only be used in a dedicated Power BI Premium workspace."

"It supports custom functions as well as parameters."

## DAX

Language developed for data analysis. It enables the creation of the following objects using expressions:

- Measures**
- Calculated Columns**
- Calculated Tables**

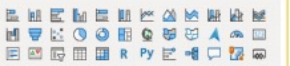
Each expression starts with the = sign, followed by links to tables/columns/functions/measures and operators. The following operators are supported:

- Arithmetic** (+, -, \*, /, ^, %)
- Comparison** (=, >, <, >=, <=, <>)
- Text concatenation** (&, &&, ||, IN)
- Precedence** ((), {})

Operators and functions require that all values/columns used are of the same data type or of a type that can be freely converted, such as a date or a number.

## Visualization

Visualizations or visuals let you present data in various graphical forms, from graphs to tables, maps, and values. Some visuals are linked to other services outside Power BI, such as Power Apps.



In addition to basic visuals, Power BI supports creating custom visuals. Custom visuals can be added using a file import or from a free Marketplace offering certified and non-certified visuals. Certification is optional, but it verifies whether, among other things, a visual accesses external services and resources.

## Themes

Serves as a single location for configuring all native graphical settings for visuals and pages.

By default, you can choose from 19 predefined themes. Custom themes can be added.

A custom theme can be applied in two different ways:

- Modification of an existing theme**—A native window that lets you modify a theme directly in the Power BI environment.
- Importing a JSON file**—Any file you create only defines the formatting that should change. Everything else remains the same. The advantage of this approach is that you can customize any single visual.

"The resulting theme can be exported in the JSON format and used in any report without the need to create a theme from scratch."

## Drill Down

The Visual that supports the embedding of hierarchies enables drilling down to the embedded hierarchy's individual levels using the following symbols:

- Drill up to a higher-level hierarchy
- Drill down to a specific field
- Drill down to the next level in the hierarchy
- Expand next-level hierarchy

## Tooltip/Custom Tooltip

**Tooltip**—A default detail preview page which appears above a visual when you hover over its values.

**Custom Tooltip**—A custom tooltip is a custom-designed report page identified as descriptive. When you hover over visual, a page appears with content filtered based on criteria specified by the value in the visual.

## Drill-through

**Drill-through** lets you pass from a data overview visual to a page with specific details. The target page is displayed with all the applied filters affecting the value from which the drill-through originated.

257,589	Display as table	100,385	12
16	Drill-through	134	140
358,125	Copy	910	

## Bookmarks

**Bookmarks** capture the currently configured view or a report page visual. Later, you can go back to that state by selecting the saved bookmark. Setting options:

- Data**—Stores filters, applied sort order in visuals and slicers. By selecting the bookmark, you can re-apply the corresponding settings.
- Display**—Stores the state of the display for visuals and report elements (buttons, images, etc.). By selecting the bookmark, you can go back to the previously stored state of the display.
- Current page**—Stores the currently displayed page. By selecting the bookmark, you can re-apply the corresponding settings.

## License

**Per-user license**

- Free**—Can be obtained for any Microsoft work or school email account. Intended for personal use. Users with this license can only use the personal workspace. They cannot share or consume shared content.
- "It is not available in 'personal workspace'"
- Pro**—It is associated with a work/school account priced at €3.40 per month or it is included in the E5 license. Intended for team collaboration. Let's users access team workspaces, consume shared content, and use apps.
- Premium per user**—Includes all Power BI Pro license capabilities, and adds features such as paginated reports, AI, greater frequency for refresh rate, XMLA endpoint, and other capabilities that are only available to Premium subscribers.

**Per-tenant license**

- Premium**—Premium is set up for individual workspaces. 0 to N workspaces can be used with a single version of the license. It provides dedicated server computing power based on license type: P1, P2, P3, P4, P5\*. It offers more space for datasets, extended metrics for individual workspaces, managed consumption of dedicated capacity, linking of Azure AI

# **SPONSORS**



# Sponsors

Fellowwind

capacity

unit<sup>it</sup>

Initiate



[stoltze][it]



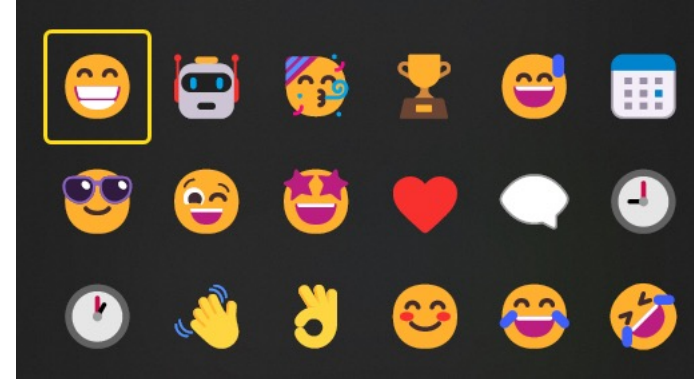
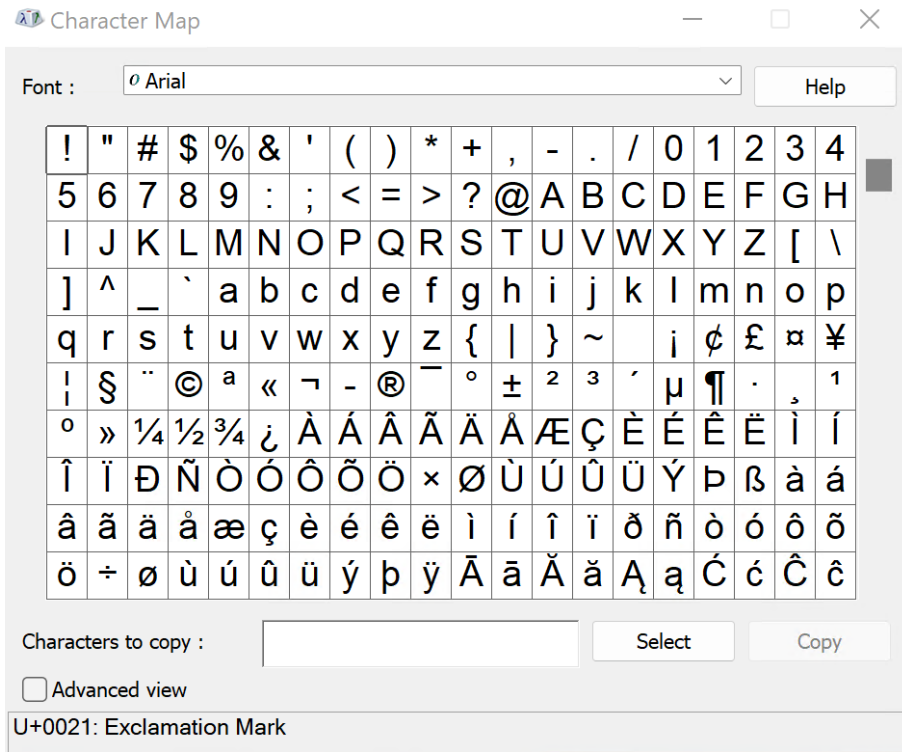
# AGENDA

- 
- **QUICK RECAP OF THE BASICS**
  - **SHOWING DATA LABELS BY PARAMETER**
  - **DYNAMIC RATIO OF TOPN**
  - **CUSTOM MULTI-COLOR GRADIENT**

# **QUICK RECAP OF THE BASICS**



Power BI can display almost any text input, including emojis.



SHORTCUT: WIN + .



# COLORS DEFINITION

HEX color code = „#RRGGBB“ & <opacity> - Opacity = {00, 80 , FF}  
„#FFE5E5FF“

HSLA color code = hsla( <hue value>, <saturation in %>, <lightness in %>, <opacity> ) - Opacity = {0, 0.5, 1}  
„hsla(0, 100%, 80%, 1)“

RGBA color code = rgba(<red>, <green>, <blue>, <opacity>) - Opacity = {0, 0.5, 1}  
„rgba(255,229,229,1)“

By Name = „red“, „grey“, „blue“,...

**Average**

**49.56**

**Average**

**51.43**

**Average**

**49.56**



# SVG ARE ALSO SUPPORTED







Requested mark:

```
data:image/svg+xml;utf8,  
data:image/svg+xml;base64,
```

Add one of this mark by M or by DAX next to SVG code:

SVGs =

```
"data:image/svg+xml;utf8,,"  
&  
SELECTEDVALUE ( '98 Power Platform'[svg] )
```

Name	Measure
anxiety.svg	
positive.svg	
PowerApps	
PowerAutomate	
PowerBI	
PowerVirtualAgents	
Total	

# FORMAT STRINGS

FORMAT( <value> , <format\_string> [, <locale\_name>] )

DEFINE

VAR N = 1 955 365 923

EVALUATE

```
{  
  ("Integer Number", FORMAT ( N, "$#,0" )),  
  ("Thousand", , FORMAT ( N, "$#,0,.0#K" )),  
  ("Million", , FORMAT ( N, "$#,0,,.0#M" )),  
  ("Billion", , FORMAT ( N, "$#,0,,,,.0#B" )),  
  ("Trillion", , FORMAT ( N, "$#,0,,,,,.0#T" )),  
}
```

FORMAT	en-US
Integer Number	\$1,955,365,923
Thousand	\$1,955,365.92K
Million	\$1,955.37M
Billion	\$1.96B
Trillion	\$0.0T

Locale_name	Jazyk	Locale_name	Jazyk
ar-SA	العربية (Arabic)	ko-KR	한국의 (Korean)
bg-BG	български (Bulgarian)	lt-LT	Lietuvos (Lithuanian)
ca-ES	català (Catalan)	lv-LV	Latvijas (Latvian)
cs-CZ	čeština (Czech)	ms-MY	Bahasa Melayu (Malay)
da-DK	dansk (Danish)	nb-NO	norsk (Norwegian)
de-DE	Deutsche (German)	nl-NL	Nederlands (Dutch)
el-GR	ελληνικά (Greek)	pl-PL	polski (Polish)
en-US	English (English)	pt-BR	português (Portuguese)
es-ES	español service (Spanish)	pt-PT	português (Portuguese)
et-EE	eesti (Estonian)	ro-RO	românesc (Romanian)
eu-ES	Euskal (Basque)	ru-RU	русский (Russian)
fi-FI	suomi (Finnish)	sk-SK	slovenský (Slovak)
fr-FR	français (French)	sl-SI	slovenski (Slovenian)
gl-ES	galego (Galician)	sr-Cyrl-RS	српски (Serbian)
he-IL	עברית (Hebrew)	sr-Latn-RS	srpski (Serbian)
hi-IN	हिन्दी (Hindi)	sv-SE	svenska (Swedish)
hr-HR	hrvatski (Croatian)	th-TH	ไทย (Thai)
hu-HU	magyar (Hungarian)	tr-TR	Türk (Turkish)
id-ID	Bahasa Indonesia (Indonesian)	uk-UA	український (Ukrainian)
it-IT	italiano (Italian)	vi-VN	tiếng Việt (Vietnamese)
ja-JP	日本の (Japanese)	zh-CN	中国 (Chinese-Simplified)
kk-KZ	Қазақ (Kazakh)	zh-TW	中國 (Chinese-Traditional)

# FORMAT STRINGS

```
FORMAT( 100, "💰;🤪;😡" )  
FORMAT(-100, "💰;🤪;😡" )  
FORMAT( 0, "💰;🤪;😡" )
```



Name	Format Date	Format	mmmm yyyy
Home table	Measures	\$ % , .00 0	Auto
Structure		Formatting	



# DYNAMIC FORMAT STRING?

## Measures

By function FORMAT()  
Set each separately  
Native Support  
Output will be TEXT  
Easier to understand within  
Self-Service

## Calculation Groups

By FORMAT STRING EXPRESSION  
Re-usable at will  
External tool is required  
Respects data type  
More complex to understand within  
Self-Service

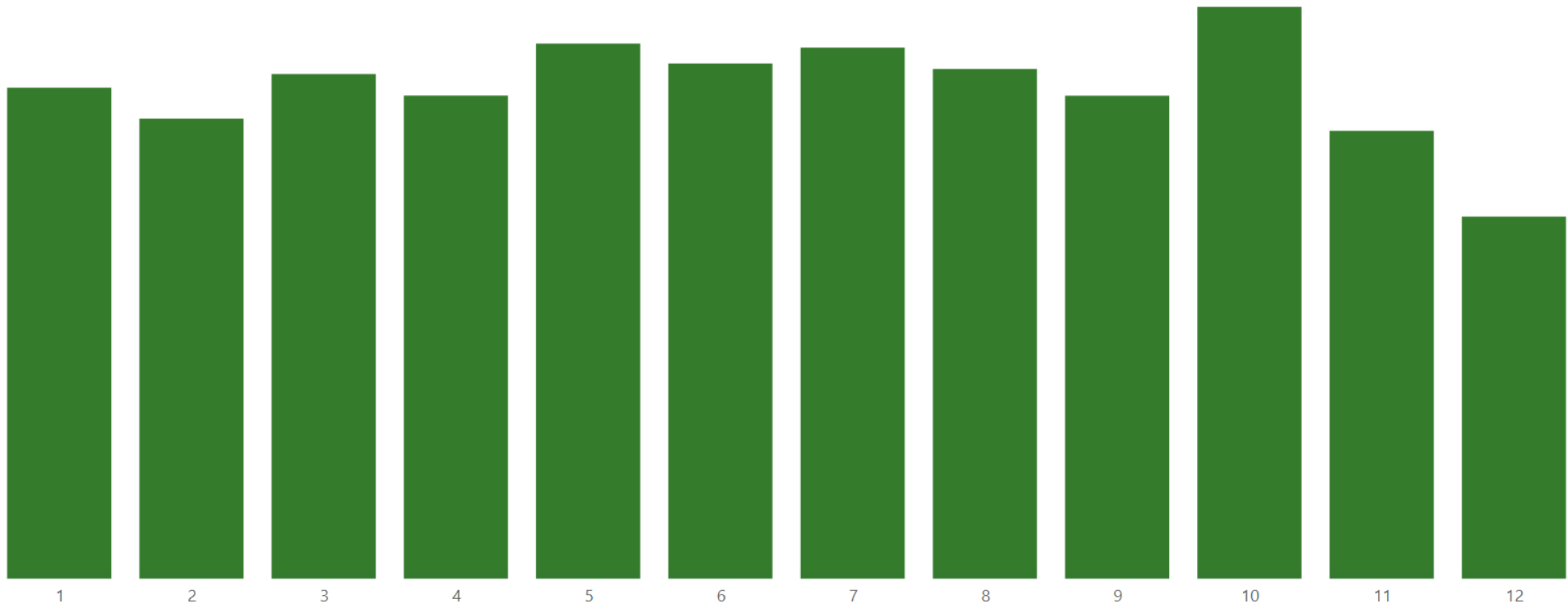




# **SHOWING DATA LABELS BY PARAMETER**

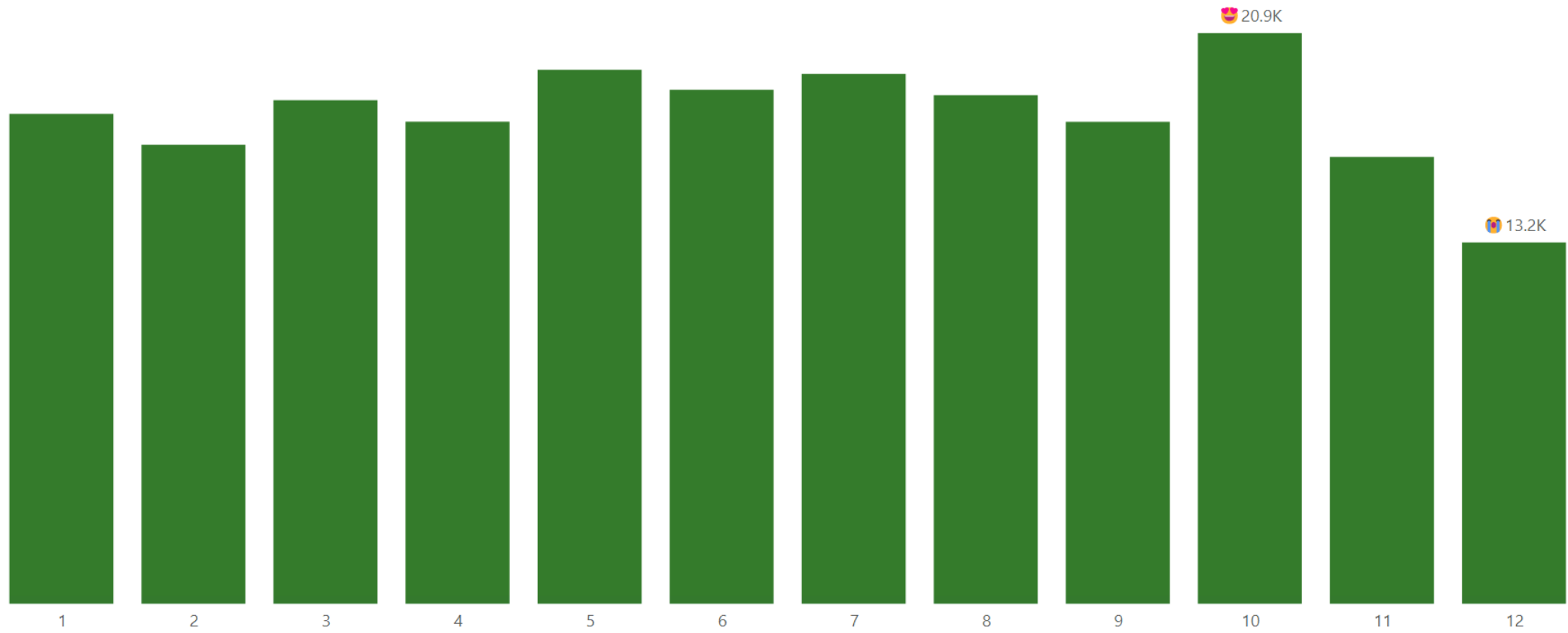
# HIGHEST AND LOWEST VALUE?

Development of the number of sales in months



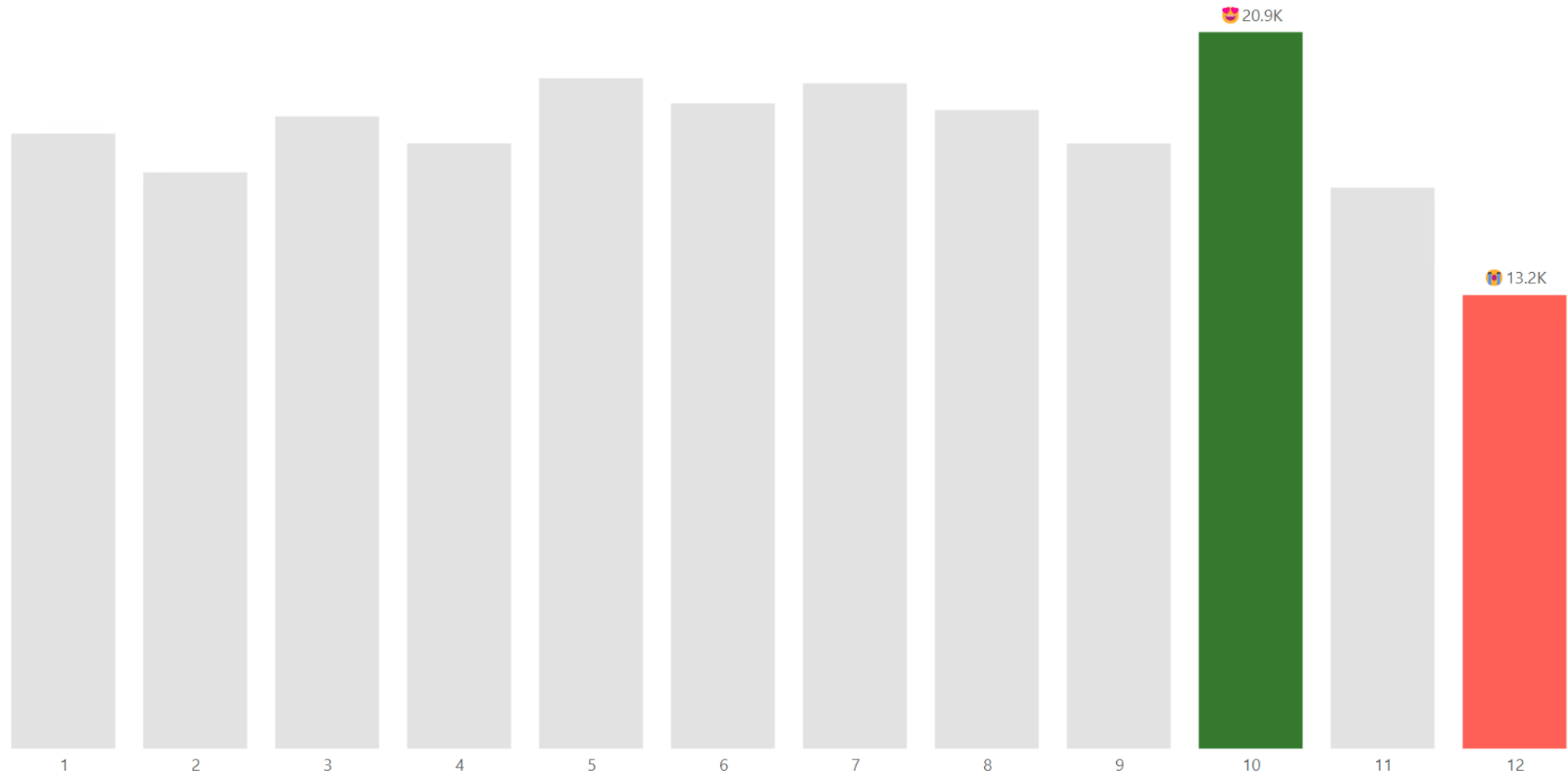
# ADDING THE LABELS

Development of the number of sales in months



# IN COMBINATION WITH COLORS

Development of the number of sales in months



# DAX for COLOR MEASURE

```
barColorChanger (Highest and Lowest) =  
VAR _monthRange =  
    ALLSELECTED ( dateKey[Month] )  
VAR _max =  
    MAXX ( _monthRange, [# Sum of Quantity] )  
VAR _min =  
    MINX ( _monthRange, [# Sum of Quantity] )  
VAR _actState = [# Sum of Quantity]  
RETURN  
    SWITCH (  
        TRUE (),  
        _actState = _max, [baseGreenColor],  
        _actState = _min, [baseRedColor],  
        [baseGrey]  
    )
```



# DAX for CALCULATED ITEM

Calculation Item - Expression (Highest and Lowest) =  
`SELECTEDMEASURE ()`

Calculation Item - Format Measure String (Highest and Lowest) =  
`VAR _dateFilter =  
 ALLSELECTED ( DateKey[Month] )  
VAR _highest =  
 MAXX ( _dateFilter, SELECTEDMEASURE () )  
VAR _lowest =  
 MINX ( _dateFilter, SELECTEDMEASURE () )  
VAR _selectedMeasure =  
 SELECTEDMEASURE ()  
RETURN  
 SWITCH (   
 TRUE (),  
 _selectedMeasure = _highest, "😍" & SELECTEDMEASUREFORMATSTRING (),  
 _selectedMeasure = _lowest, "😞" & SELECTEDMEASUREFORMATSTRING (),  
 UNICHAR ( 8203 )  
 )`



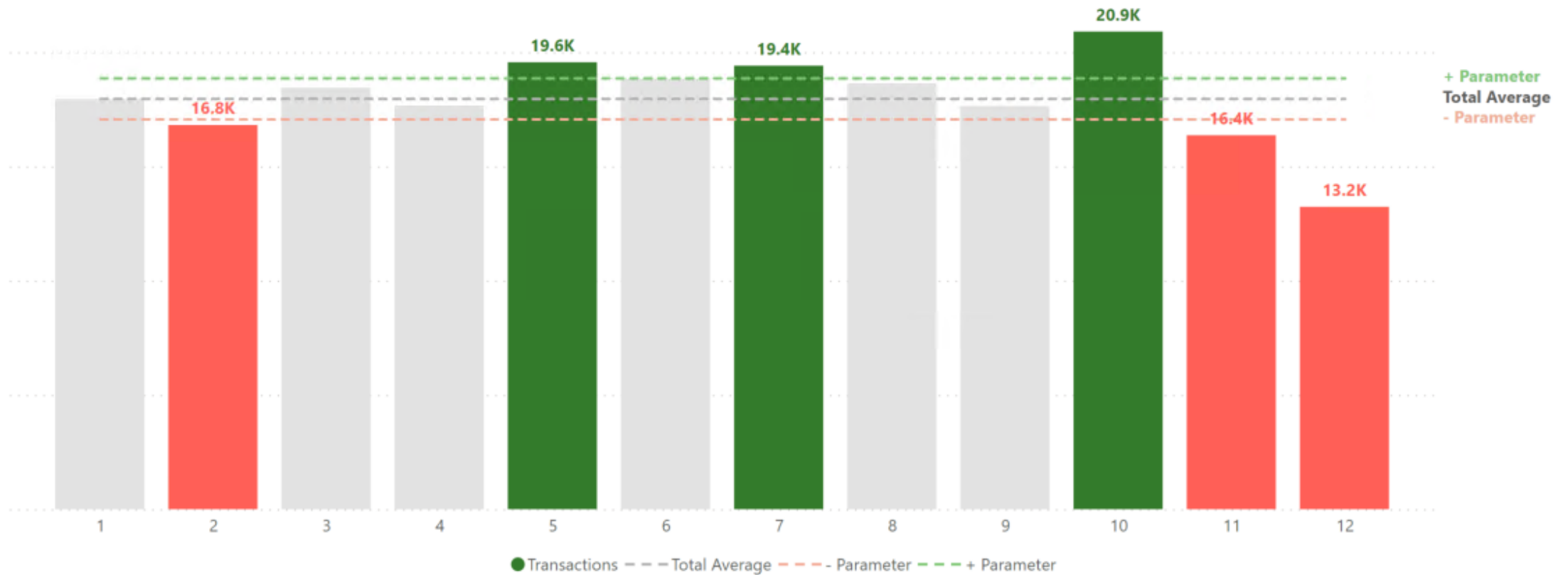


# LABELS FOR MONTHS OUTSIDE DEVIATION

Months that fall outside the deviation

Deviation %:

5



# DAX for COLOR MEASURE

```
colorOfBarsByParameter =  
VAR _avgUp = [# Average (+Procentage)]  
VAR _avgDown = [# Average (-Procentage)]  
VAR _value = [# Sum of Quantity]  
RETURN  
    SWITCH (  
        TRUE (),  
        _value > _avgUp, "#347B2B",  
        _value < _avgDown, "#FF5F57",  
        "#E2E2E2"  
    )
```



# DAX for CALCULATED ITEM

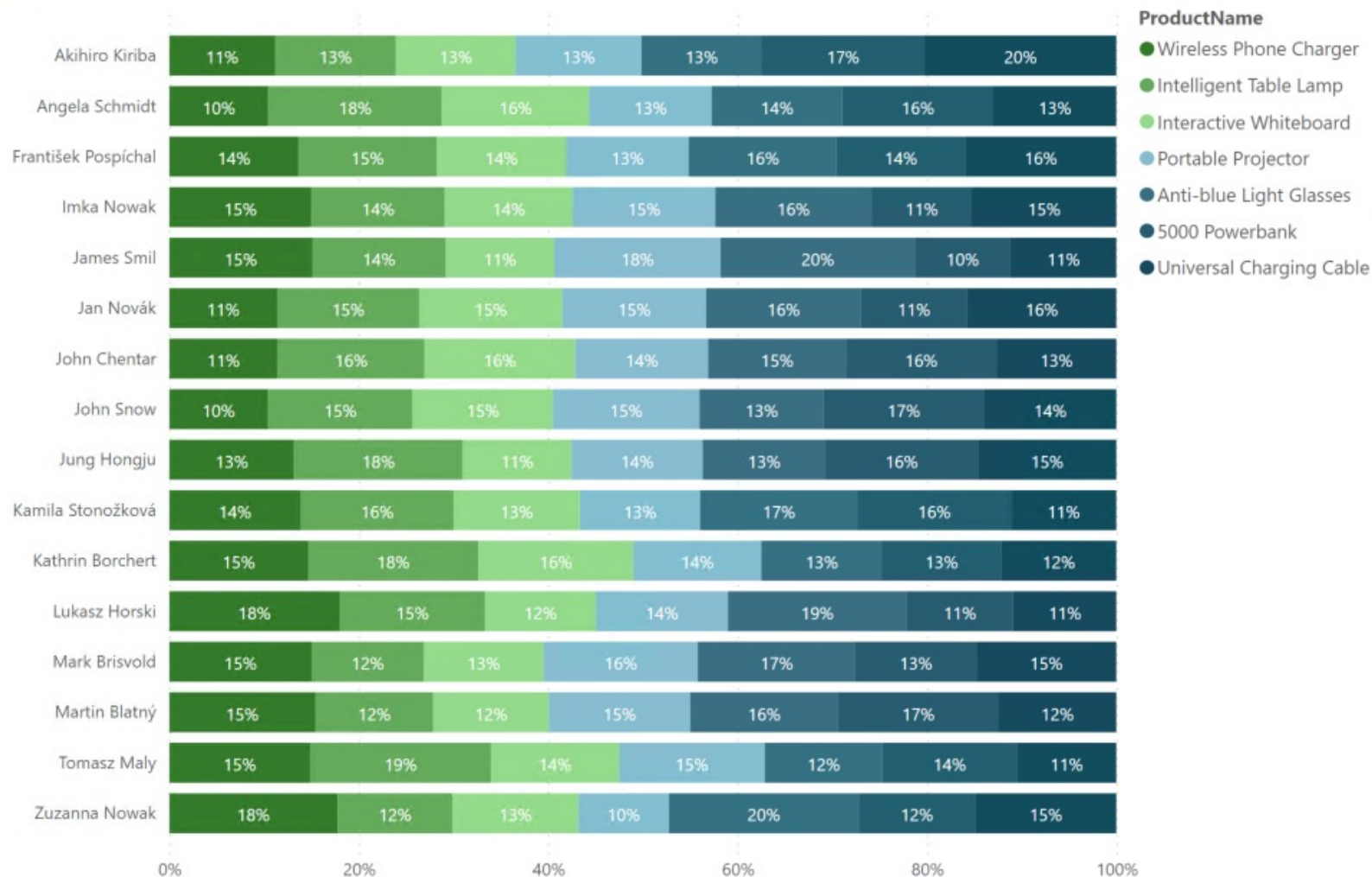
```
ShowDataLabelsByParameter (Calculation Item) =
SWITCH (
    TRUE (),
    ISSELECTEDMEASURE (
        [# Average Of Total],
        [# Average (+Procentage)],
        [# Average (-Procentage)]
    ),
    VAR _maxOfRange =
        MAXX ( ALLSELECTED ( DateKey[Month] ), DateKey[Month] )
    VAR _selectedMonth =
        SELECTEDVALUE ( DateKey[Month] )
    RETURN
        IF (
            _maxOfRange = _selectedMonth,
            SELECTEDMEASUREFORMATSTRING (),
            UNICHAR ( 8203 )
        ),
    ISSELECTEDMEASURE ( [# Sum of Quantity] ),
    VAR _avgUp = [# Average (+Procentage)]
    VAR _avgDown = [# Average (-Procentage)]
    VAR _value =
        SELECTEDMEASURE ()
    RETURN
        IF (
            _value > _avgUp
            || _value < _avgDown,
            SELECTEDMEASUREFORMATSTRING (),
            UNICHAR ( 8203 )
        ),
    SELECTEDMEASUREFORMATSTRING ()
)
```



# **DYNAMIC RATIO OF TOPN**

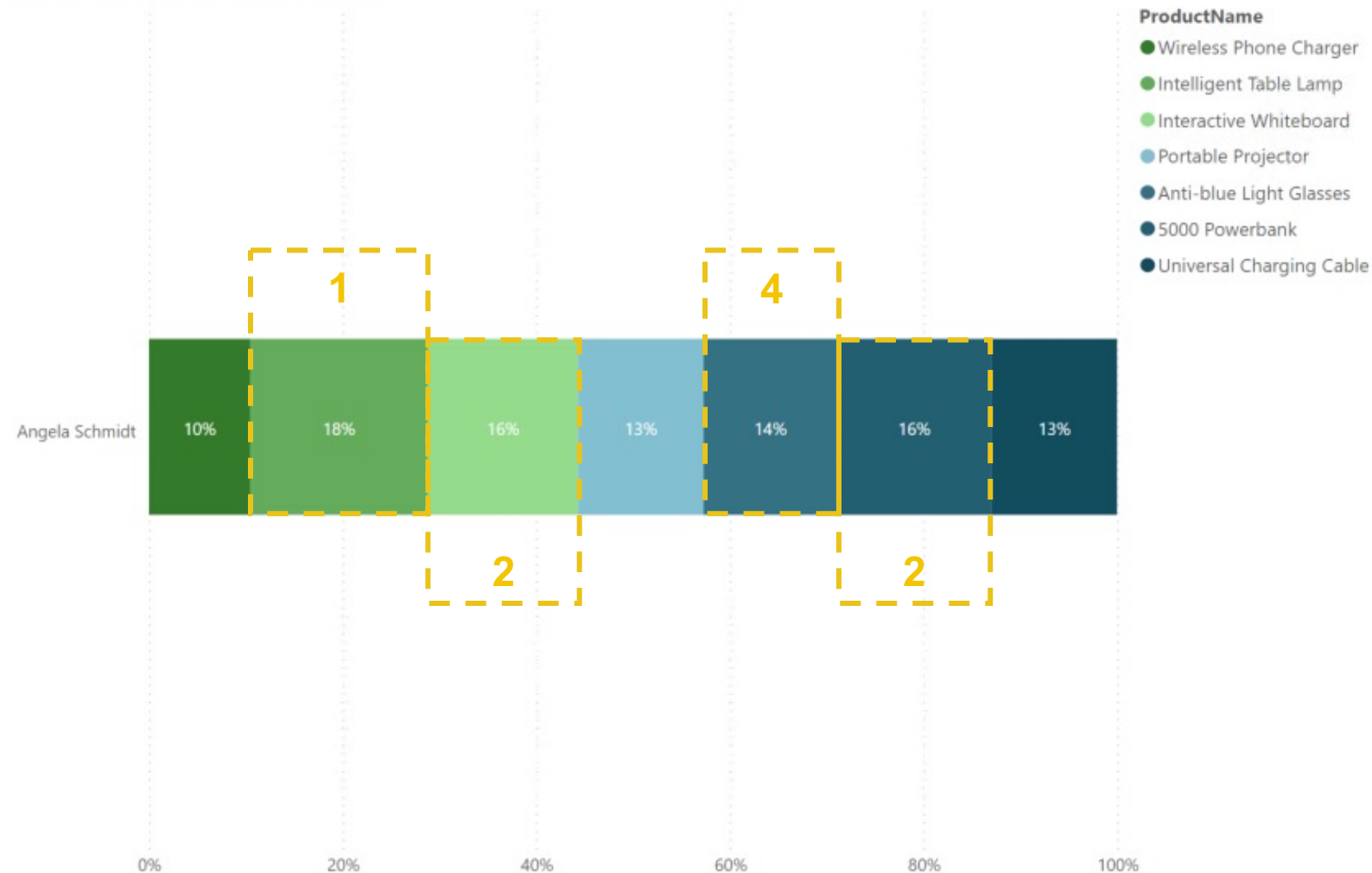
# RATIO OF SOLD PRODUCTS BY SALESMAN

Ratio of sold products by salesman



# TOP FOUR SOLD PRODUCTS BY ANGELA SCHMIDT

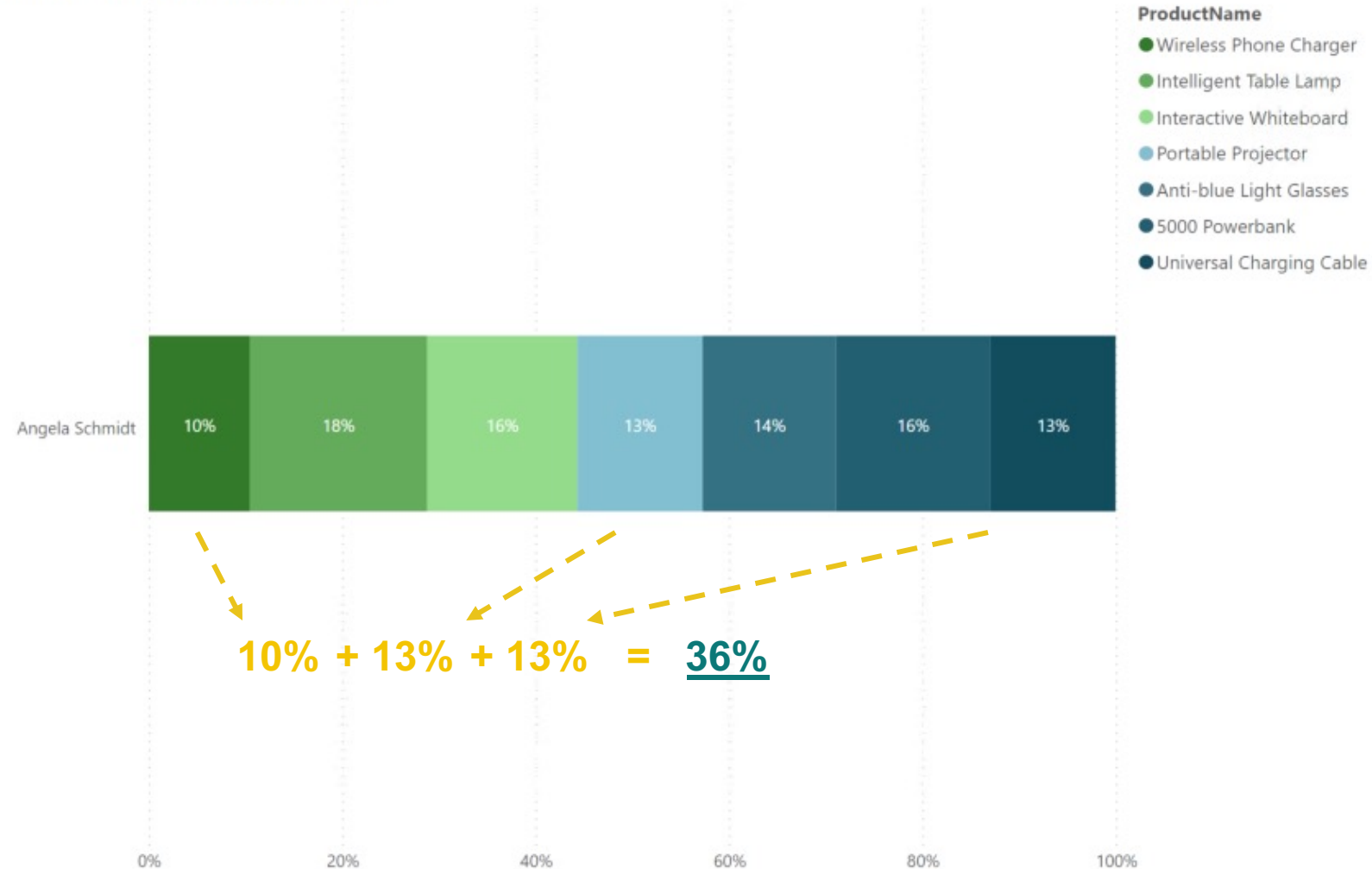
Ratio of sold products by salesman





# HOW MUCH % ARE ALL OTHER PRODUCTS?

Ratio of sold products by salesman



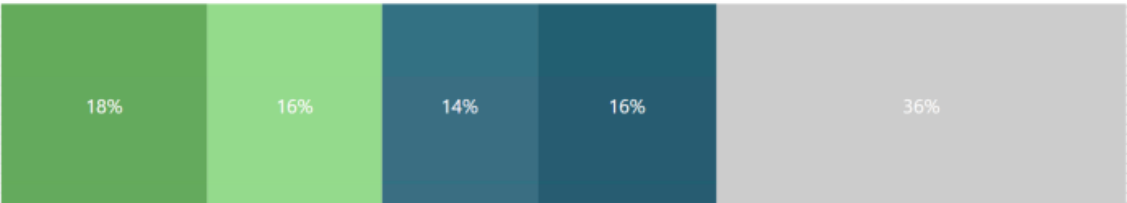
# WHAT ABOUT NOW?

Dynamic ratio by TOPN

TOP PRODUCTS BY SALESMAN: 4

- Products
- Intelligent Table Lamp
  - Interactive Whiteboard
  - Anti-blue Light Glasses
  - 5000 Powerbank
  - Other

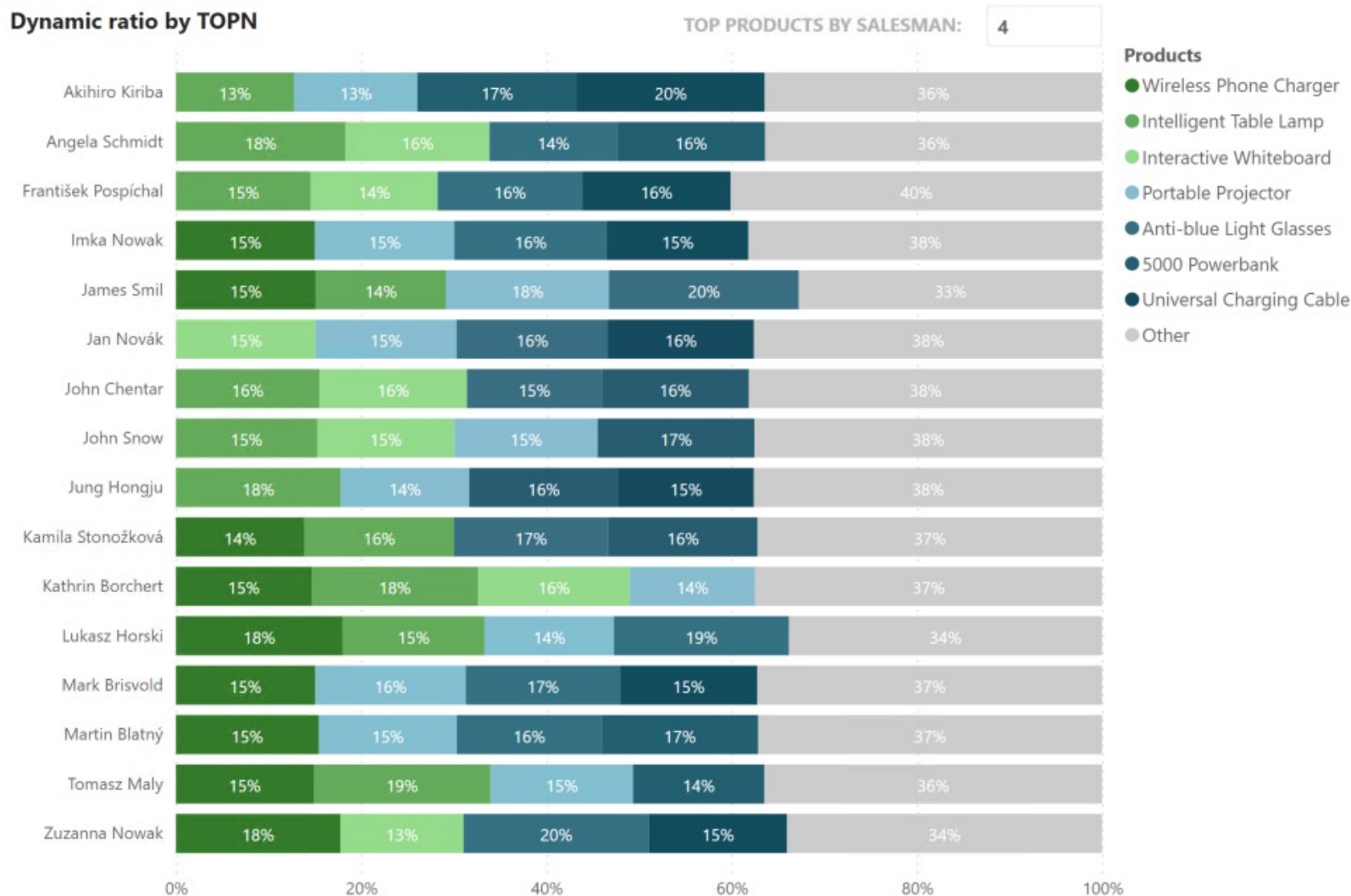
Angela Schmidt



0% 20% 40% 60% 80% 100%



# DYNAMIC RATIO OF SOLD PRODUCTS BY SALESMAN



## MEASURE FOR DYNAMIC RATIO

Dynamic Ratio of TOPN Sales =

```
VAR _nProducts = [Threshold Value]
VAR _rankingGroup =
    SELECTEDVALUE ( TableOfProduct[label] )
VAR _topProducts =
    TOPN ( _nProducts, ALLSELECTED ( TableOfProduct ), [# Sum of Quantity] )
RETURN
    SWITCH (
        _rankingGroup,
        "products", CALCULATE ( [# Sum of Quantity], KEEPFILTERS ( _topProducts ) ),
        "others",
            VAR _topCalculated =
                CALCULATE ( [# Sum of Quantity], _topProducts )
            VAR _allAmount =
                CALCULATE ( [# Sum of Quantity], ALLSELECTED ( TableOfProduct ) )
            VAR _result = _allAmount - _topCalculated
            RETURN
                _result
    )
```



# **CUSTOM MULTI-COLOR GRADIENT**



# GRADIENT IN POWER BI

FocusCategory	Quantity Sold
[-] <b>Non-Priority</b>	<b>295434</b>
Anti-blue Light Glasses	79800
Portable Projector	72861
5000 Powerbank	71857
Universal Charging Cable	70916
[+] <b>Priority</b>	<b>218416</b>
Intelligent Table Lamp	77726
Wireless Phone Charger	71107
Interactive Whiteboard	69583
<b>Total</b>	<b>513850</b>

ProductName	Quantity Sold
Anti-blue Light Glasses	79800
Intelligent Table Lamp	77726
Portable Projector	72861
5000 Powerbank	71857
Wireless Phone Charger	71107
Universal Charging Cable	70916
Interactive Whiteboard	69583
<b>Total</b>	<b>513850</b>

Lightness of  
color is  
decreasing



FocusCategory	Quantity Sold
[-] <b>Non-Priority</b>	<b>295434</b>
Anti-blue Light Glasses	79800
Portable Projector	72861
5000 Powerbank	71857
Universal Charging Cable	70916
[+] <b>Priority</b>	<b>218416</b>
Intelligent Table Lamp	77726
Wireless Phone Charger	71107
Interactive Whiteboard	69583
<b>Total</b>	<b>513850</b>

ProductName	Quantity Sold
Anti-blue Light Glasses	79800
Intelligent Table Lamp	77726
Portable Projector	72861
5000 Powerbank	71857
Wireless Phone Charger	71107
Universal Charging Cable	70916
Interactive Whiteboard	69583
<b>Total</b>	<b>513850</b>

Background color  
respects its  
category, and they  
are simultaneously  
decreasing its  
lightness





# RESPECTING ROW GRANULARITY

Employee	Wireless Phone Charger	Intelligent Table Lamp	Interactive Whiteboard	Portable Projector	Anti-blue Light Glasses	5000 Powerbank	Universal Charging Cable
Akihiro Kiriba	3954	4523	4471	4714	4478	6092	7178
Angela Schmidt	3335	5845	4955	4129	4399	5082	4147
František Pospíchal	4514	4814	4532	4271	5162	4479	5288
Imka Nowak	4832	4531	4350	4843	5303	3410	4916
James Smil	4848	4520	3678	5641	6575	3243	3594
Jan Novák	3368	4428	4460	4493	4823	3310	4663
John Chentar	3681	4990	5118	4511	4679	5102	4080
John Snow	3058	4475	4346	4534	3843	4959	4106
Jung Hongju	4512	6098	3928	4744	4432	5503	5014
Kamila Stonožková	4467	5193	4282	4092	5351	5196	3594
Kathrin Borchert	4615	5660	5153	4257	4010	3950	3841
Lukasz Horski	5582	4761	3623	4330	5850	3488	3383
Mark Brisvold	5501	4332	4631	5957	6091	4663	5419
Martin Blatný	4649	3759	3679	4501	4713	5100	3760
Tomasz Maly	4746	6080	4308	4910	3962	4510	3362
Zuzanna Nowak	5445	3717	4069	2934	6129	3770	4571
<b>Total</b>	<b>71107</b>	<b>77726</b>	<b>69583</b>	<b>72861</b>	<b>79800</b>	<b>71857</b>	<b>70916</b>

Every row have own gradient context



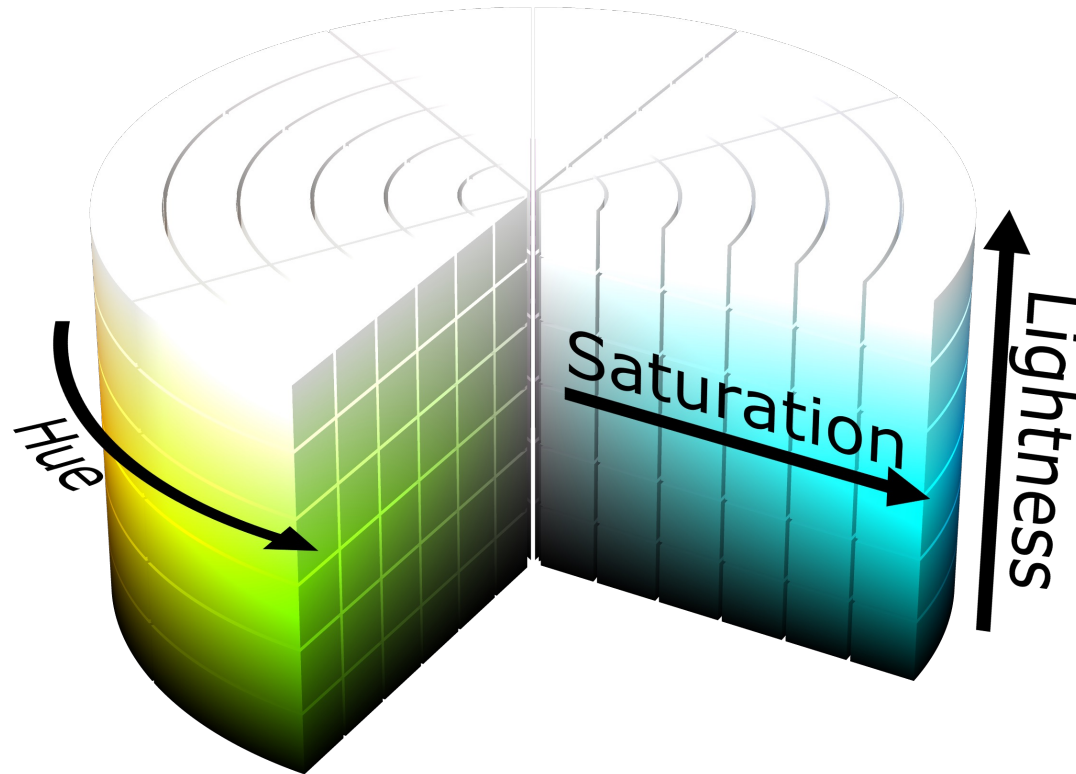
# GRADIENT ON COLUMN GRANULARITY

Employee	Wireless Phone Charger	Intelligent Table Lamp	Interactive Whiteboard	Portable Projector	Anti-blue Light Glasses	5000 Powerbank	Universal Charging Cable
Akihiro Kiriba	3954	4523	4471	4714	4478	6092	7178
Angela Schmidt	3335	5845	4955	4129	4399	5082	4147
František Pospíchal	4514	4814	4532	4271	5162	4479	5288
Imka Nowak	4832	4531	4350	4843	5303	3410	4916
James Smil	4848	4520	3678	5641	6575	3243	3594
Jan Novák	3368	4428	4460	4493	4823	3310	4663
John Chentar	3681	4990	5118	4511	4679	5102	4080
John Snow	3058	4475	4346	4534	3843	4959	4106
Jung Hongju	4512	6098	3928	4744	4432	5503	5014
Kamila Stonožková	4467	5193	4282	4092	5351	5196	3594
Kathrin Borchert	4615	5660	5153	4257	4010	3950	3841
Lukasz Horski	5582	4761	3623	4330	5850	3488	3383
Mark Brisvold	5501	4332	4631	5957	6091	4663	5419
Martin Blatný	4649	3759	3679	4501	4713	5100	3760
Tomasz Maly	4746	6080	4308	4910	3962	4510	3362
Zuzanna Nowak	5445	3717	4069	2934	6129	3770	4571
<b>Total</b>	<b>71107</b>	<b>77726</b>	<b>69583</b>	<b>72861</b>	<b>79800</b>	<b>71857</b>	<b>70916</b>

Same gradient context for all rows



# HSL COLOR



HSL color code = `hsl( <hue value>, <saturation in %>, <lightness in %> )`  
„`hsl(0, 100%, 80%)`“



# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "##%" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
```



# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =  
VAR _selectedValue =  
    SELECTEDVALUE ( 'priceList'[Id] )  
VAR _selectedCategory =  
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )  
VAR _inputTable =  
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )  
VAR _counter =  
    COUNTROWS ( _inputTable ) - 1  
VAR _rankOfRow =  
    MINX (  
        FILTER (  
            ADDCOLUMNS (  
                _inputTable,  
                "@rank",  
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1  
            ),  
            [Id] = _selectedValue  
        ),  
        [@rank]  
    )  
VAR _lightness =  
    VAR _initialLightness = 0.33  
    VAR _maxLightness = 1  
    VAR _lightnessMover =  
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )  
    RETURN  
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "##%" )  
VAR _green =  
    VAR _hue = "113"  
    VAR _saturation = "48%"  
    RETURN  
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"  
VAR _blue =  
    VAR _hue = "194"  
    VAR _saturation = "54%"  
    RETURN  
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"  
RETURN  
    SWITCH (  
        TRUE (),  
        _selectedCategory = "Priority", _green,  
        _selectedCategory = "Non-Priority", _blue,  
        BLANK ()  
    )
```

id	ProductName	UnitPrice	ProductionPrice	Margin	FocusCategory
1	Wireless Phone Charger	13	8	0.625	Priority
2	Intelligent Table Lamp	73	50	0.46	Priority
3	Interactive Whiteboard	1245	450	1.76666666666667	Priority
4	Portable Projector	150	45	2.33333333333333	Non-Priority
5	Anti-blue Light Glasses	100	20	4	Non-Priority
6	5000 Powerbank	23	6	2.83333333333333	Non-Priority
7	Universal Charging Cable	10	3	2.33333333333333	Non-Priority

# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "##%" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
```

id	ProductName	UnitPrice	ProductionPrice	Margin	FocusCategory
1	Wireless Phone Charger	13	8	0.625	Priority
2	Intelligent Table Lamp	73	50	0.46	Priority
3	Interactive Whiteboard	1245	450	1.76666666666667	Priority
4	Portable Projector	150	45	2.33333333333333	Non-Priority
5	Anti-blue Light Glasses	100	20	4	Non-Priority
6	5000 Powerbank	23	6	2.83333333333333	Non-Priority
7	Universal Charging Cable	10	3	2.33333333333333	Non-Priority



# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "##" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
)
```


id	ProductName	UnitPrice	ProductionPrice	Margin	FocusCategory
1	Wireless Phone Charger	13	8	0.625	Priority
2	Intelligent Table Lamp	73	50	0.46	Priority
3	Interactive Whiteboard	1245	450	1.76666666666667	Priority
4	Portable Projector	150	45	2.33333333333333	Non-Priority
5	Anti-blue Light Glasses	100	20	4	Non-Priority
6	5000 Powerbank	23	6	2.83333333333333	Non-Priority
7	Universal Charging Cable	10	3	2.33333333333333	Non-Priority

id
1
2
3
4
5
6
7



# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "##" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
```



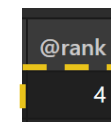
id	@rank
1	4
2	1
3	6
4	2
5	0
6	3
7	5





# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "#%" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
)
```



$$1) 1 - 0.33 \div ( 0.67 / 7 ) \doteq 0.1117$$

$$2) 0.33 + ( 0.1117 * 4 ) \doteq 0.7766$$

$$3) 77.66\% \rightarrow 78\%$$



# CUSTOM MULTI-COLOR GRADIENT

```

DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
    FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "0.##" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
    "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
    "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )

```

Base colors definitions



# CUSTOM MULTI-COLOR GRADIENT

```
DynamicMultiColorGradient =
VAR _selectedValue =
    SELECTEDVALUE ( 'priceList'[Id] )
VAR _selectedCategory =
    SELECTEDVALUE ( 'pricelist'[FocusCategory] )
VAR _inputTable =
    CALCULATETABLE ( DISTINCT ( 'priceList'[id] ), ALLSELECTED ( 'priceList' ) )
VAR _counter =
    COUNTROWS ( _inputTable ) - 1
VAR _rankOfRow =
    MINX (
        FILTER (
            ADDCOLUMNS (
                _inputTable,
                "@rank",
                RANKX ( ALLSELECTED ( 'priceList' ), CALCULATE ( [# Sum Of Quantity] ),, DESC ) - 1
            ),
            [Id] = _selectedValue
        ),
        [@rank]
    )
VAR _lightness =
    VAR _initialLightness = 0.33
    VAR _maxLightness = 1
    VAR _lightnessMover =
        DIVIDE ( ( _maxLightness - _initialLightness ), _counter )
    RETURN
        FORMAT ( _initialLightness + ( _lightnessMover * _rankOfRow ), "%%" )
VAR _green =
    VAR _hue = "113"
    VAR _saturation = "48%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
VAR _blue =
    VAR _hue = "194"
    VAR _saturation = "54%"
    RETURN
        "hsl(" & _hue & "," & _saturation & "," & _lightness & ")"
RETURN
    SWITCH (
        TRUE (),
        _selectedCategory = "Priority", _green,
        _selectedCategory = "Non-Priority", _blue,
        BLANK ()
    )
)
```

ProductName	Quantity Sold	DynamicMultiColorGradient
Anti-blue Light Glasses	79800	hsl(194,54%,33%)
Intelligent Table Lamp	77726	hsl(113,48%,44%)
Portable Projector	72861	hsl(194,54%,55%)
5000 Powerbank	71857	hsl(194,54%,67%)
Wireless Phone Charger	71107	hsl(113,48%,78%)
Universal Charging Cable	70916	hsl(194,54%,89%)
Interactive Whiteboard	69583	hsl(113,48%,100%)
<b>Total</b>	<b>513850</b>	

hsl(113,48%,78%)



**This sample is just a fraction of what can be done with DAX in Power BI. Therefore, it is important not to put up with someone saying that "something" is not working and to try to solve it yourself.**



**Q&A TIME**

**THANK YOU FOR THE ATTENTION**



**AND A SPECIAL THANKS ALSO GOES  
TO THE ENTIRE ORGANIZING TEAM**