Comprehensive DAX Calculations Using Power BI for Retail and Healthcare Data

Document Overview:

This document presents a detailed analysis of sales data for a retail company, utilizing the Adventure Works dataset, along with an in-depth examination of patient wait times and total patient visits in a hospital setting. Additionally, it includes various other measures that were explored while studying DAX (Data Analysis Expressions) comprehensively. The analysis has been performed using Power BI, employing a range of DAX (Data Analysis Expressions) calculations to derive key insights. This document provides detailed descriptions of the DAX measures used and their relevance to the sales analysis.

Content:

DAX on Sales Analysis project

To calculate the percentage of all orders

```
% of All Orders = DIVIDE([Total Orders],[All Orders])
```

To calculate the percentage of all returns

```
% of All Returns = DIVIDE([Total Returns], [All Returns])
```

To calculate 10 days rolling revenue

```
10-day Rolling Revenue = CALCULATE([Total Revenue],
   DATESINPERIOD('Calendar Lookup'[Date],
      MAX('Calendar Lookup'[Date]),-10,DAY))
```

To calculate adjusted price

```
Adjusted Price = [Average Retail Price] * (1 + 'Price Adjustments (%)'[Price Adjustments (%) Value])
```

To calculate adjusted profit

```
Adjusted Profit = [Adjusted Revenue] - [Total Cost]
```

To calculate adjusted revenue

```
Adjusted Revenue = SUMX('Sales Data', 'Sales Data'[OrderQuantity] * [Adjusted Price])
```

To calculate total orders in sales data

```
All Orders = CALCULATE([Total Orders], ALL('Sales Data'))
```

To calculate total returns in sales data

```
All Returns = CALCULATE([Total Returns], ALL('Returns Data'))
```

To calculate the average of the retail price

```
Average Retail Price = AVERAGE('Product Lookup'[ProductPrice])
```

To calculate average revenue per customer

```
Average Revenue Per Customer = [Total Revenue] / [Total Customer]
```

To calculate the return rate of the bike

```
Bike Return Rate = CALCULATE([Return Rate],
   'Product Categories Lookup'[CategoryName] = "Bikes")
```

Comprehensive DAX Calculations Using Power BI for Retail and Healthcare Data

To calculate total bike returns

```
Bike Returns = CALCULATE([Total Returns],
   'Product Categories Lookup'[CategoryName] = "Bikes")
```

To calculate bike sales

```
Bike Sales = CALCULATE([Quantity Sold],'Product Categories Lookup'[CategoryName] = "Bikes")
```

To calculate bulk orders

```
Bulk Orders = CALCULATE([Total Orders],'Sales Data'[Order Quantity] > 1,
   FILTER('Sales Data', 'Sales Data'[Order Quantity] > 1))
```

To calculate product with higher price when compared with overall average price

```
High Ticker Orders = CALCULATE([Total Orders], FILTER('Product Lookup','Product Lookup'[ProductPrice] > [Overall Average Price]))
```

To calculate Target

```
Order Target = [Previous Month Orders] * 1.1
```

To calculate target gap

```
Order Target Gap = [Total Orders] - [Order Target]
```

To calculate overall average retail price for products

```
Overall Average Price = CALCULATE([Average Retail Price],ALL('Product Lookup'))
```

To calculate previous month orders

```
Previous Month Orders = CALCULATE ([Total Orders], DATEADD (‘Calendar Lookup'[Date], -1,
      MONTH))
```

To calculate previous month profit

```
Previous Month Profit = CALCULATE ([Total Profit], DATEADD('Calendar Lookup'[Date], -1,
      MONTH))
```

To calculate previous month returns

```
Previous Month Returns = CALCULATE ([Total Returns], DATEADD('Calendar Lookup'[Date],
      -1, MONTH))
```

To calculate previous month revenue

```
Previous Month Revenue = CALCULATE ([Total Revenue], DATEADD('Calendar Lookup'[Date],
      -1, MONTH))
```

To calculate profit target

```
Profit Target = [Previous Month Profit] * 1.1
```

To calculate profit target gap

Comprehensive DAX Calculations Using Power BI for Retail and Healthcare Data

Profit Target Gap = [Total Profit] - [Profit Target]

To calculate sum of return quantity
Quantity Return = SUM('Returns Data'[ReturnQuantity])

To calculate quantity sold
Quantity Sold = SUM('Sales Data'[OrderQuantity])

To calculate return rate
Return Rate = [Quantity Return] / [Quantity Sold]

To calculate revenue target
Revenue Target = [Previous Month Revenue] * 1.1

To calculate revenue target gap
Revenue Target Gap = [Total Revenue] - [Revenue Target]

To calculate total cost
Total Cost = SUMX('Sales Data', 'Sales Data'[Order Quantity] * RELATED('Product Lookup'[ProductCost]))

To calculate total distinct customer
Total Customer = DISTINCTCOUNT('Sales Data'[CustomerKey])

To calculate total distinct orders
Total Orders = DISTINCTCOUNT('Sales Data'[OrderNumber])

To calculate total orders using customer details
Total Orders (Customer Detail) = IF(HASONEVALUE('Customer Lookup'[CustomerKey]), [Total Orders], "-")

To calculate total profit
Total Profit = [Total Revenue] - [Total Cost]

To calculate total returns
Total Returns = COUNT ('Returns Data'[ReturnQuantity])

To calculate total revenue
Total Revenue = SUMX('Sales Data', 'Sales Data'[OrderQuantity] * RELATED('Product Lookup'[ProductPrice]))

To calculate total revenue using customer details
Total Revenue (Customer Detail) = IF(HASONEVALUE('Customer Lookup'[CustomerKey]), [Total Revenue], "-")

To calculate weekend order
Weekend Orders = CALCULATE([Total Orders], 'Calendar Lookup'[Weekend] = "Weekend")

To calculate YTD revenue

YTD Revenue = CALCULATE([Total Revenue], DATESYTD('Calendar Lookup'[Date]))

****

DAX on patient wait time project

To calculate the percentage of the admission scheduled

```
% Admin Schedule = DIVIDE(COUNTROWS(FILTER('Patient Details',
'Patient Details'[patient_admin_flag]=TRUE())), [Total Patients])
```

To calculate the percentage of total number of female visits

```
% Female Visits = DIVIDE(CALCULATE([Total Patients],'Patient Details'[patient_gender]="F"), [Total Patients])
```

To calculate the percentage of total number of male visits

```
% Male Visits = DIVIDE(CALCULATE([Total Patients], 'Patient Details'[patient_gender]="M"),
        [Total Patients])
```

To calculate the percentage with no rating at all

```
% No Rating =
    VAR _NoRatings = CALCULATE([Total Patients],
                     'Patient Details'[patient_sat_score] = BLANK())
    RETURN
    DIVIDE(_NoRatings, [Total Patients])
```

To calculate the percentage with no admission scheduled

```
% Non-Admin Schedule = DIVIDE(
                        COUNTROWS(
                            FILTER(
                                'Patient Details',
                                'Patient Details'[patient_admin_flag]=FALSE())
                        ),
                        [Total Patients]
                    )
```

To calculate the percentage of patients with no referral

```
% Non-Referred Patients =
VAR _FilterPatients =
     CALCULATE([Total Patients], 'Patient Details'[department_referral] = "none")
     RETURN
     DIVIDE(_FilterPatients, [Total Patients])
```

To calculate the percentage of referred pateints

```
% Referred Patients =
VAR _FilterPatients =
```

```
    CALCULATE([Total Patients], 'Patient Details'[department_referral]<> "none")
    RETURN
    DIVIDE(_FilterPatients, [Total Patients])
```

To calculate the percentage of patients in unknown category

```
% Unknown =
    DIVIDE(CALCULATE([Total Patients], 'Patient Details'[patient_gender]="NC"),
       [Total Patients])
```

To calculate average satisfaction score

```
Average Satisfaction Score = CALCULATE (AVERAGE('Patient Details'[patient_sat_score]),'Patient
Details'[patient_sat_score]<>BLANK())
```

To calculate average wait time

```
Average WaitTime = AVERAGE('Patient Details'[patient_waittime])
```

To calculate the cumulative frequency of maximum point of patient visit monthly

```
CF Max Point(Month) =
   VAR _PatientTable =
       CALCULATETABLE(
        ADDCOLUMNS(
        SUMMARIZE('Date', 'Date'[Month]),
        "@TotalPatients", [Total Patients]
        ),
        ALLSELECTED()
       )
    VAR _MinValue = MINX(_PatientTable, [@TotalPatients])
    VAR _MaxValue = MAXX(_PatientTable, [@TotalPatients])
    VAR _TotalPatients =[Total Patients]
    RETURN
    SWITCH(
        TRUE(),
        _TotalPatients = _MinValue,0,
        _TotalPatients = _MaxValue,1
    )
```

To calculate the cumulative frequency of maximum point of patient visit yearly

```
CF Max Point(Year) =
   VAR _PatientTable =
       CALCULATETABLE(
        ADDCOLUMNS(
        SUMMARIZE('Date', 'Date'[Year]),
        "@TotalPatients", [Total Patients]
        ),
```

```
        ALLSELECTED()
       )
    VAR _MinValue = MINX(_PatientTable, [@TotalPatients])
    VAR _MaxValue = MAXX(_PatientTable, [@TotalPatients])
    VAR _TotalPatients =[Total Patients]
    RETURN
    SWITCH(
        TRUE(),
        _TotalPatients = _MinValue,0,
        _TotalPatients = _MaxValue,1
    )
```

To calculate the correlation between patient satisfaction and wait time according to age group

```
Map Caption =
VAR _SelectedMeasure =
     SELECTEDVALUE(Parameter[Parameter Order])
     RETURN
     IF(_SelectedMeasure = 0,
     "Most SATISFIED patients represents the darkest GREEN on Age-Group",
     "The darkest GREEN on the scale represents LOW WAIT TIME on Age-Group"
     )
```

To calculate total patients in the table

```
Total Patients = COUNTROWS('Patient Details')
```

****

DAX measures practice problems

Calculating the total of sales using SUM

```
Total Sum of Sales = SUM(Sales[Total Revenue])
```

Calculating the count of rows in Sales table

```
Total Transaction = COUNTROWS(Sales)
```

Calculating the total cost in Sales table

```
Total Cost = SUMX(Sales, Sales[Order Quantity]*Sales[Total Unit Cost])
```

Filtering customer data with wholesales and whose name index is between 5 and 20

```
Customer Wholesales =
    CALCULATE([Total Transaction],
       FILTER(Sales, Sales[Customer Name Index] > 5 && Sales[Customer Name Index] < 20),
           FILTER(Sales, Sales[Channel] = "Wholesale") )
```

Calculating total sales average per channel

```
Total Sales average per channel =
   AVERAGEX(
```

```
    VALUES(Sales[Channel]),
        [Total Sales])
```

Categorizing Product sales more than 10M and less than 10M in Products look up table

```
Product Groups =
   SWITCH( TRUE(),
       Products[Sales] > 10000000 , "Top",
       Products[Sales] <= 10000000, "Bottom",
       BLANK())
```

Can also be written using already created measures

```
Product Groups =
   SWITCH( TRUE(),
       [Total Sales] > 10000000 , "Top",
       [Total Sales] <= 10000000, "Bottom",
       BLANK())
```

Using measures in Context Transition to calcuakte customer sales

```
Customer Sales = [Total Sales]
```

Calculating Count of Distinct products measure to filter customers and products based on scenario

```
Total Products Bought = DISTINCTCOUNT(Sales[Product Description Index])
```

Calculating Average cost using the measure already created

```
Average Cost = AVERAGEX(Sales, [Total Cost])
```

Calculating Minimum and maximum cost

```
Min costs = MINX(Sales, [Total Cost])
Max costs = MAXX(Sales, [Total Cost])
```

Calculating profit with measures created

```
Total Profits = [Total Sales] - [Total Cost]
```

Calculating profit margin using measure branching

```
Profit Margin = DIVIDE([Total Profits], [Total Sales], 0)
```

Calculating Profit Margin for sales greater than 40%

Using Calculate,

```
Sales Segment =
   CALCULATE([Total Sales],
       FILTER(Sales, [Profit Margin] > 0.40) )
```

Using iterative function,

```
Sales Segment 2 = SUMX(FILTER(Sales, [Profit Margin] > 0.40), [Total Sales])
```

Calculating Last year sales

```
Sales LY = CALCULATE([Total Sales], DATEADD(Dates[Date], -1, YEAR))
```

Error handling for blanks, calculating Year-Over-Year sales

```
Yoy Sales Diff = IF(ISBLANK([Sales LY]), BLANK(), [Total Sales] - [Sales LY] )
```