ZTM

POWER BI CHEAT SHEET

TRAVIS CUZICK

V1.01

HEEELLLOOOOO!

I'm Andrei Neagoie, Founder and Lead Instructor of the Zero To Mastery Academy.

After working as a Senior Software Developer over the years, I now dedicate 100% of my time to teaching others valuable software development skills, help them break into the tech industry, and advance their careers.

In only a few years, **over one million students** around the world have taken Zero To Mastery courses and many of them are now working at top tier companies like Apple, Google, Amazon, Tesla, IBM, Facebook, and Shopify, just to name a few.

This cheat sheet, created by our Power BI instructor (Travis Cuzick) provides you with the key Power BI concepts and information that you need to know and remember.

If you want to not only learn Power BI but also get the exact steps to build your own projects and get hired as an Analyst, then check out our Career Paths.

Happy Coding!
Andrei

Founder & Lead Instructor, Zero To Mastery
Andrei Neagoie

Power BI Cheatsheet

This Power BI cheat sheet provides a quick reference guide for beginners and advanced users, explaining the essential concepts and features, including Power BI components, data sources, data modeling, the DAX formula language, Power BI dashboards, data visualizations, Power Query, the M language, and best practices. Keep it handy while working with Power BI to improve your understanding and proficiency.

Power BI Components

Power BI Desktop

Power BI Desktop is a Windows application used for creating and designing reports and dashboards.

- Report view: Design and build visuals, arrange and format them on report pages.
- Data view: Inspect and manage tables, columns, and measures in the data model.
- Relationships view: Create, modify, and visualize relationships between tables.
- External tools: Integrate with tools like Tabular Editor, DAX Studio, and ALM Toolkit for advanced modeling and development.

Power BI Service

Power BI Service is a cloud-based platform for sharing, managing, and collaborating on reports and dashboards.

- Workspaces: Organize and collaborate on content by creating and managing workspaces.
- Data refresh: Schedule and manage data refreshes to keep reports up-to-date.
- Row-level security: Implement security rules to restrict data access for specific users.
- Sharing: Share reports, dashboards, or individual visuals with others.
- Apps: Package and distribute a collection of related content through apps.
- Subscriptions: Set up email subscriptions for reports and dashboards.

- Dataflows: Create reusable and scalable data preparation processes with Power Query in the cloud.
- Embedding: Integrate Power BI reports and dashboards into custom applications or websites.

Power BI Report Server

Power BI Report Server is an on-premise server solution for hosting and sharing reports within an organization.

- Report management: Organize, manage, and access reports through a web portal.
- Report scheduling: Schedule data refreshes and report delivery.
- Security: Implement role-based security and data access restrictions.
- Integration: Integrate with SQL Server Analysis Services (SSAS) for advanced data modeling.

Power BI Mobile

Power BI Mobile is an app for viewing and interacting with reports and dashboards on smartphones and tablets.

- Access reports and dashboards on iOS, Android, and Windows devices.
- Share snapshots of visuals through email or text messages.
- Receive data-driven alerts based on specified conditions.

Power BI Embedded

Power BI Embedded enables the integration of Power BI reports and dashboards into custom applications for seamless reporting capabilities.

- Embedding: Integrate Power BI visuals, reports, or dashboards into custom applications, websites, or portals.
- APIs: Use Power BI REST APIs to interact with and manage Power BI content programmatically.
- Row-level security: Implement dynamic security rules based on application users.

Custom Visuals

Custom visuals extend the visualization capabilities of Power BI by providing additional chart types and features.

- Marketplace: Access and import custom visuals from the marketplace.
- Custom visuals development: Develop custom visuals using the Power BI Custom Visuals SDK.
- Security: Review custom visuals for security and compliance before using them in your reports.

Data Connectors

Power BI data connectors enable connections to a wide variety of data sources for importing and transforming data.

- Built-in connectors: Use built-in connectors for common data sources like files, databases, and cloud services.
- Custom connectors: Develop custom connectors using the Power Query M formula language and the Power Query SDK.
- Certified connectors: Access certified connectors from third-party developers in the marketplace.

Integration with Other Tools

Power BI can integrate with various tools and services to extend its capabilities and streamline workflows.

- Excel: Import Excel data, analyze Power BI data in Excel using Analyze in Excel, or use Power BI Publisher for Excel.
- Azure Synapse Analytics: Integrate with Azure Synapse for large-scale data processing and advanced analytics.
- Azure Machine Learning: Incorporate Azure Machine Learning models into Power BI for predictive analytics.
- Power Apps: Embed Power Apps into Power BI reports for a seamless, interactive experience.

- Power Automate: Use Power Automate to trigger actions or workflows based on Power BI data, alerts, or user interactions.
- Power Query: Utilize Power Query, available in Power BI and Excel, for data transformation and preparation.
- SQL Server Analysis Services (SSAS): Connect to SSAS for advanced data modeling and analytics.
- Dynamics 365: Integrate with Dynamics 365 applications for seamless data analysis and reporting.
- Microsoft Teams: Embed Power BI reports and dashboards in Microsoft Teams for easy access and collaboration.
- SharePoint: Embed Power BI reports in SharePoint Online using the Power BI web part.

Power BI Development

Develop custom solutions, automate processes, and extend Power BI capabilities using various development tools.

- Power BI REST APIs: Use REST APIs for programmatic access to Power BI resources like datasets, reports, and dashboards.
- Power BI JavaScript API: Utilize the JavaScript API for embedding and interacting with Power BI content in custom applications.
- Power BI Custom Visuals SDK: Develop custom visuals using the Power BI Custom Visuals SDK and TypeScript.
- Power Query M formula language: Write custom functions, expressions, and transformations using the Power Query M formula language.
- DAX: Leverage Data Analysis Expressions (DAX) for creating custom calculations, measures, and KPIs in Power BI data models.

Licensing

Understand and choose the appropriate Power BI licenses for you or your organization's needs.

- Power BI Free: Access basic features of Power BI Desktop for individual report creation and analysis.

- Power BI Pro: Obtain a cloud-based per-user license for creating, sharing, and collaborating on reports and dashboards.
- Power BI Premium: Get dedicated cloud or on-premise capacity for large-scale deployments, advanced features, and improved performance.
- Power BI Premium per User: Access Premium features with a per-user license for small and medium-sized organizations.
- Power BI Report Server: Use a separate, on-premise server solution for hosting and sharing reports within an organization.

Data Sources

Power BI can connect to various data sources, enabling you to work with diverse types of data.

Files

Import data from various file formats to use in Power BI.

- Excel: Connect to Excel workbooks (.xls, .xlsx, .xlsm) and import data from sheets, tables, or named ranges.
- CSV: Import data from comma-separated values (.csv) files.
- XML: Connect to XML (.xml) files and import data using XML table inference.
- JSON: Connect to JSON (.json) files and import data using JSON document structure.
- PDF: Extract tables from PDF (.pdf) files for data analysis.
- SharePoint Folder: Access files stored in a SharePoint folder and import data.

Databases

Connect to various on-premise and cloud-based databases.

- SQL Server: Connect to SQL Server databases and import tables, views, or write custom SQL queries.
- Azure SQL Database: Access Azure SQL Database tables, views, or custom SQL queries.

- Azure Synapse Analytics: Query data stored in Azure Synapse Analytics (formerly Azure SQL Data Warehouse).
- MySQL: Connect to MySQL databases and import tables, views, or custom SQL queries.
- PostgreSQL: Access PostgreSQL databases and import tables, views, or custom SQL queries.
- Oracle: Connect to Oracle databases and import tables, views, or custom SQL queries.
- IBM Db2: Access IBM Db2 databases and import tables, views, or custom SQL queries.
- SAP HANA: Connect to SAP HANA databases and import tables, views, or custom SQL queries.
- Teradata: Access Teradata databases and import tables, views, or custom SQL queries.

Cloud Services

Connect to various cloud-based services for data analysis.

- Azure Blob Storage: Import data from files stored in Azure Blob Storage.
- Azure Data Lake Storage: Access data stored in Azure Data Lake Storage Gen1 and Gen2.
- Azure Table Storage: Connect to Azure Table Storage and import data from tables.
- Azure Cosmos DB: Access data stored in Azure Cosmos DB using SQL API or Gremlin API.
- Amazon Redshift: Connect to Amazon Redshift databases and import tables, views, or custom SQL queries.
- Google BigQuery: Access data stored in Google BigQuery tables or write custom SQL queries.
- Snowflake: Connect to Snowflake databases and import tables, views, or custom SQL queries.

Online Services

Import data from popular online services and platforms.

- Dynamics 365: Connect to Dynamics 365 applications like Sales, Finance, and Operations.
- Salesforce: Access Salesforce objects, reports, or write custom SOQL queries.
- SharePoint Online: Connect to SharePoint Online lists, libraries, or folders.
- Google Analytics: Import data from Google Analytics reports and dimensions.
- Adobe Analytics: Access Adobe Analytics data and dimensions.
- QuickBooks Online: Connect to QuickBooks Online data for financial analysis.

Connectors

Use various connectors to access data from other sources.

- OData: Connect to OData feeds and import data from OData services.
- REST APIs: Access data from REST APIs using the Web connector and custom authentication.
- Custom Connectors: Develop custom connectors using the Power Query M formula language and Power Query SDK.
- Dataflows: Import data from Power BI dataflows for reusable data preparation processes.
- R Script: Execute R scripts to import and transform data.
- Python Script: Execute Python scripts to import and transform data.
- DirectQuery: Connect to certain data sources using DirectQuery for real-time data access and analysis.

Data Integration

Combine and transform data from multiple sources.

- Merge Queries: Combine data from multiple sources based on matching columns (similar to SQL JOIN).
- Append Queries: Stack data from similar sources (similar to SQL UNION).

- Reference Queries: Create a new query that references an existing query, allowing for data transformations without duplicating data.
- Custom Functions: Develop custom functions in Power Query M formula language to apply complex transformations to multiple sources.

Data Refresh

Configure data refresh settings to keep Power BI reports up-to-date.

- Import Mode: Schedule data refreshes for imported datasets in the Power BI Service.
- DirectQuery Mode: Access data in real-time for supported data sources without the need for scheduled refreshes.
- Incremental Refresh: Configure incremental refresh policies to load only new or changed data for large datasets.
- On-Premises Data Gateway: Install and configure the on-premises data gateway to enable data refresh for on-premise data sources.
- Data Gateway Scheduling: Set up data refresh schedules for datasets using the on-premises data gateway.

Security and Authentication

Configure security and authentication settings for data sources.

- Anonymous: Connect to data sources without providing credentials.
- Basic: Use username and password for data source authentication.
- API Key: Provide an API key for authentication with certain online services and APIs.
- OAuth2: Use OAuth2 authentication for secure access to online services like Salesforce or SharePoint Online.
- Database: Utilize database-specific authentication methods like Windows, SQL Server, or Oracle.
- Organizational Account: Authenticate with a Microsoft work or school account for services like Dynamics 365 or SharePoint Online.

- Custom: Implement custom authentication methods for specific data sources or APIs.

Data Modeling

Data modeling helps you shape, clean, and structure data to create meaningful reports.

Importing Data

Start by connecting to various data sources and importing data into Power BI.

- Get Data: Home tab > Get Data to connect to data sources like Files, Databases, Cloud Services, or Connectors.

Query Editor

Power BI's Query Editor is the key tool for shaping, cleaning, and transforming data.

- Open Query Editor: Home tab > Transform Data or double-click the query in the 'Fields' pane.

Navigation and Queries Management

Navigate and manage queries efficiently in the Query Editor.

- Queries pane: View and manage all queries in the data model.
- Applied Steps: View, modify, or delete the transformation steps applied to a query.
- Properties: Rename the query, add a description, or enable/disable data load.

Data Transformation

Apply various transformations to clean, shape, and structure your data.

- Remove columns: Right-click column header > Remove or Home tab > Remove Columns.
- Rename columns: Double-click the column header.
- Reorder columns: Drag and drop column headers.
- Change data type: Click the data type icon on the column header and select the appropriate data type.
- Split columns: Home tab > Split Column to divide a column into multiple columns based on a delimiter or character count.

- Merge columns: Add Column tab > Merge Columns to combine two or more columns into one.
- Fill down/up: Right-click column header > Fill Down or Fill Up to fill empty cells with adjacent values.
- Replace values: Right-click column header > Replace Values to find and replace specific values.
- Group by: Transform tab > Group By to aggregate data based on specific criteria.
- Pivot and Unpivot: Transform tab > Pivot or Unpivot to change the data structure.

Advanced Data Transformation

Apply advanced transformations to handle complex data modeling scenarios.

- Conditional column: Add Column tab > Conditional Column to create a new column based on conditions.
- Custom column: Add Column tab > Custom Column to create a new column using custom expressions.
- Merging queries: Home tab > Merge Queries to combine data from multiple sources based on matching columns.
- Appending queries: Home tab > Append Queries to stack data from similar sources.
- Parameters: Home tab > Manage Parameters to create dynamic data models.
- Functions: Create custom functions using the Advanced Editor or M language.

Data Modeling and Relationships

Build a functional data model by creating relationships between tables.

- Star schema: Aim to create a star schema with a central fact table and multiple dimension tables.
- Relationships view: Use the Relationships view to create, modify, and visualize relationships between tables.
- Create relationships: Drag and drop fields between tables in the Relationships view.
- Edit relationships: Double-click the relationship line to modify the relationship.
- Cross-filter direction: Set filtering direction (Both or Single) in relationships.

- Relationship types: Define One-to-One, One-to-Many, Many-to-One, or Many-to-Many relationships.

Calculated Columns and Measures

Leverage DAX (Data Analysis Expressions) to create custom calculations.

- Calculated columns: Create new columns using DAX expressions. Syntax: `NewColumnName = DAX_Expression`
- Measures: Create custom measures using DAX expressions. Syntax: `NewMeasure = DAX_Expression`
- Basic DAX functions: SUM, COUNT, AVERAGE, MIN, MAX, RELATED, IF, DISTINCTCOUNT, RANKX
- Time intelligence functions: SAMEPERIODLASTYEAR, TOTALYTD, DATEADD
- Filter functions: CALCULATE, FILTER, ALL, ALLEXCEPT

Data Model Optimization

Optimize your data model for better performance and faster report loading.

- Remove unnecessary columns: Remove columns that are not needed for analysis to reduce data model size and improve performance.
- Data compression: Use data types with smaller storage sizes (e.g., Int64 instead of Decimal) for better compression.
- Sort by column: Use the "Sort by Column" feature to improve sorting performance in visuals.
- Use measures: Use measures instead of calculated columns for better performance in large datasets.
- Optimize DAX: Write efficient DAX expressions to reduce calculation times.
- Bi-directional relationships: Use bi-directional relationships cautiously, as they may cause performance issues and ambiguous results.
- Incremental refresh: Implement incremental refresh to reduce data load times for large datasets.

Best Practices

Follow best practices to create efficient and effective data models in Power BI.

- Understand your data: Familiarize yourself with the data sources and structure before starting data modeling.
- Plan your data model: Define your goals, requirements, and data relationships before building the data model.
- Keep it simple: Aim for simplicity and maintainability in your data model.
- Naming conventions: Use clear and consistent naming conventions for tables, columns, and measures.
- Use hierarchies: Create hierarchies to facilitate drill-down and drill-up capabilities in visuals.
- Error handling: Implement error handling techniques like IFERROR and DIVIDE to prevent calculation errors.
- Document your work: Maintain documentation for complex calculations and measures to help other users understand your data model.

Power BI Dashboards

Dashboard Creation

Create interactive and visually appealing dashboards to display key insights and KPIs.

- Pin Visuals: Pin visuals from reports to an existing or new dashboard.
- Dashboard Tiles: Arrange and resize tiles to optimize dashboard layout and design.
- Tile Interactions: Enable or disable interactions between dashboard tiles.
- Dashboard Themes: Apply pre-built or custom themes to enhance dashboard appearance.

Dashboard Navigation

Organize and navigate through dashboards for an efficient user experience.

- Dashboard List: Access and manage dashboards through the Power BI Service dashboard list.
- Favorites: Mark frequently used dashboards as favorites for easy access.
- Workspace Navigation: Navigate between dashboards within a workspace.

- Full-Screen Mode: View dashboards in full-screen mode for presentations or focused analysis.

Dashboard Interactivity

Provide interactive features to engage users and enable data exploration.

- Dashboard Slicers: Use slicers to filter data across multiple visuals in a dashboard.
- Drill-down: Implement drill-down capabilities to explore data at different levels of granularity.
- Tooltips: Customize tooltips to display additional details when users hover over visuals.
- Cross-Filtering: Enable cross-filtering between visuals for a seamless, interactive experience.

Real-Time Dashboards

Display real-time data in dashboards to monitor KPIs and trends as they occur.

- Streaming Datasets: Create streaming datasets to push real-time data to Power BI.
- DirectQuery: Connect to certain data sources using DirectQuery for real-time data access and analysis.
- Automatic Page Refresh: Set up automatic page refresh intervals for visuals displaying real-time data.

Data-Driven Alerts

Configure alerts to notify users when specific data conditions are met.

- Alert Rules: Set up rules based on data values or conditions for individual visuals.
- Notification Channels: Receive alerts through email or Power BI Mobile app notifications.
- Alert Management: View, manage, and edit existing alerts in the Power BI Service.

Dashboard Sharing

Share dashboards with colleagues and stakeholders for collaboration and decision-making.

- Share Dashboards: Share dashboards with individual users, groups, or the entire organization.
- Publish to Web: Publish dashboards to the web for public access (not recommended for sensitive data).
- Embed Dashboards: Embed dashboards into custom applications, websites, or portals using Power BI Embedded.
- Dashboard Permissions: Set permissions for shared dashboards, allowing users to view, edit, or manage content.

Dashboard Performance

Optimize dashboard performance to ensure a smooth and responsive user experience.

- Data Model Optimization: Optimize data models by removing unnecessary columns, applying data compression, and using measures.
- Visual Optimization: Choose efficient visual types and limit the number of visuals on a single dashboard.
- Query Optimization: Write efficient DAX expressions and optimize data queries for faster data retrieval.

Mobile Dashboards

Design mobile-friendly dashboards for easy access and interaction on smartphones and tablets.

- Mobile Layout: Create a separate mobile layout for dashboards with optimized visuals and navigation.
- Power BI Mobile App: Access and interact with dashboards on iOS, Android, and Windows devices.
- Mobile Features: Utilize mobile-specific features like geolocation, data-driven alerts, and snapshot sharing.

Dashboard Best Practices

Implement best practices to create effective and user-friendly dashboards.

- Clear Objectives: Define the purpose and objectives of each dashboard before designing it.

- Consistent Design: Maintain a consistent design across dashboards using themes, colors, and visual styles.
- Focus on Key Metrics: Display only the most relevant KPIs and insights for the target audience.
- Data Visualization Guidelines: Follow data visualization best practices to ensure accurate and clear representation of data.
- Accessibility: Design dashboards with accessibility in mind, using clear fonts, contrasting colors, and descriptive tooltips for users with disabilities.

Dashboard Maintenance

Regularly maintain and update dashboards to ensure data accuracy and relevance.

- Data Refresh: Schedule data refreshes for imported datasets or use DirectQuery for real-time data access.
- Dashboard Versioning: Use version control to track changes and maintain a history of dashboard updates.
- Dashboard Review: Periodically review dashboards to ensure they still meet business objectives and user needs.
- User Feedback: Collect user feedback and incorporate suggestions for dashboard improvements.

Power BI Data Visualization

Visualization Basics

Understand the fundamentals of data visualization in Power BI for effective communication and data storytelling.

- Visual Types: Choose from a wide range of built-in visualizations or use custom visuals from the marketplace.
- Visual Interactions: Configure interactions between visuals, such as cross-filtering and highlighting.
- Report Canvas: Design and organize visuals on the report canvas, using grids and guides for alignment.

- Themes: Apply built-in or custom themes to control the appearance and style of visuals consistently.

Core Visuals

Leverage core Power BI visuals to display data effectively and efficiently.

- Bar and Column Charts: Use vertical or horizontal bars to compare categorical data across categories.
- Line and Area Charts: Display trends over time or continuous data using lines or filled areas.
- Pie and Donut Charts: Represent proportions of a whole using pie or donut charts, but use cautiously due to perceptual limitations.
- Scatter and Bubble Charts: Plot data points on an X-Y plane to visualize relationships between two or three measures.
- Tables and Matrix: Display tabular data with sorting, conditional formatting, and grouping capabilities.
- Cards and Multi-row Cards: Present single data points or KPIs using cards or multi-row cards.
- Gauges and KPIs: Show progress toward a goal or target using gauges or KPI visuals.

Advanced Visuals

Utilize advanced Power BI visuals for more complex data representation and analysis.

- Treemap: Visualize hierarchical data or part-to-whole relationships using nested rectangles.
- Ribbon Chart: Display ranked data over time, with ribbons connecting data points to highlight ranking changes.
- Waterfall Chart: Show the cumulative effect of sequential positive and negative values.
- Histogram: Visualize the distribution of data across continuous intervals or categories.
- Heatmap: Use color intensity to represent data values in a matrix or table.

- Map Visuals: Display geographical data using various map visuals, such as ArcGIS Maps, Shape Maps, or Filled Maps.

Custom Visuals

Extend Power BI capabilities with custom visuals from the marketplace or develop your own.

- Marketplace Visuals: Browse the Power BI marketplace to find and add custom visuals to your reports.
- Certified Visuals: Use certified visuals that have been verified by Microsoft for quality and security.
- Custom Visual Development: Develop custom visuals using TypeScript and the Power BI Custom Visual SDK.

Visual Formatting

Apply formatting options to enhance the appearance and readability of visuals.

- Data Colors: Customize the color palette for data series or data points.
- Data Labels: Display data values or labels directly on visuals for better clarity.
- Title, Legend, and Axis: Format titles, legends, and axes to improve readability and understanding.
- Gridlines and Data Table: Add gridlines or data tables to charts for reference and comparison.

Visual Analytics

Enhance data analysis and exploration with visual analytics features.

- Tooltips: Add custom tooltips to display additional information or context for data points.
- Drill-down: Enable drill-down functionality to explore hierarchical data in more detail.
- Forecasting: Use built-in forecasting to project future trends based on historical data.
- Conditional Formatting: Apply conditional formatting to tables, matrices, or charts to highlight data points based on rules or conditions.

Mobile Reporting

Design and optimize reports for mobile devices and different screen sizes.

- Mobile Layout: Create a mobile layout for reports with visuals optimized for smaller screens.
- Responsive Visuals: Use responsive visuals that automatically adapt to different screen sizes and orientations.
- Mobile Interactions: Ensure visual interactions and navigation are intuitive and easy to use on mobile devices.

Exporting and Sharing

Export and share Power BI visuals and reports with various audiences and formats.

- Export to PowerPoint, PDF, or CSV: Export visuals and reports to PowerPoint, PDF, or CSV formats for sharing and distribution.
- Publish to Web: Publish reports to the web for public access, but consider data privacy and security.
- Sharing within Power BI: Share reports and dashboards with other Power BI users within your organization.
- Power BI Embedded: Embed Power BI visuals and reports in web applications or other platforms using the Power BI Embedded service.

Visual Storytelling Techniques

Employ storytelling techniques to create engaging and impactful Power BI reports.

- Narrative Structure: Organize your report with a clear beginning, middle, and end to guide users through the analysis.
- Context: Provide context and background information to help users understand the data and its significance.
- Focus: Emphasize key insights or takeaways using visual hierarchy, color, and annotations.
- Interactivity: Encourage users to explore the data with interactive visuals and filtering options.

Performance Optimization

Optimize Power BI visual performance for faster rendering and smoother user experience.

- Limit Data Points: Reduce the number of data points in visuals to improve rendering performance.
- Visual Aggregations: Use visual-level aggregations or summary data when possible to reduce query complexity.
- Incremental Rendering: Apply incremental rendering techniques, such as paging or lazy loading, for large datasets.

Design Principles

Apply fundamental design principles to create visually appealing and effective Power BI reports.

- Visual Hierarchy: Establish a visual hierarchy to guide users' attention to the most important elements.
- Balance and Alignment: Maintain balance and alignment in the layout and arrangement of visuals.
- Typography: Use consistent and legible typography to improve readability and understanding.
- White Space: Utilize white space effectively to separate and group visual elements without clutter.

Data Visualization Tools and Integrations

Leverage additional tools and integrations to enhance your Power BI data visualization capabilities.

- R and Python Visuals: Create custom visuals using R or Python scripts for advanced data visualization and analysis.
- Custom Connectors: Develop custom connectors to access additional data sources for visualization.
- Third-Party Integrations: Integrate Power BI with other tools, such as Excel, Tableau, or Qlik, for additional visualization and analysis options.

Data Visualization Best Practices

Implement data visualization best practices for effective communication and storytelling.

- Choose the Right Visual: Select the most appropriate visual type to represent the data and answer specific questions.
- Less is More: Avoid clutter and unnecessary elements to focus on the most important information.
- Color Theory: Use colors strategically to draw attention, encode data, and establish hierarchy.
- Consistency: Maintain consistent formatting and design across visuals and reports.
- Accessibility: Design visuals that are accessible to users with disabilities, including colorblindness and screen reader compatibility.

Power BI DAX

DAX Basics

Understand the fundamentals of Data Analysis Expressions (DAX) language for Power BI data models.

- Calculated Columns: Create new columns in data tables using DAX expressions.
- Measures: Define dynamic calculations that aggregate data based on the user's selection or filter context.
- KPIs: Develop key performance indicators to track specific business goals or targets.
- Variables: Use variables for storing intermediate results in DAX expressions.
- Data Types: Work with various data types such as integer, decimal, text, currency, datetime, and boolean.

DAX Functions

Leverage a wide range of DAX functions to perform complex calculations and data manipulation.

- Aggregation Functions: Use SUM, AVERAGE, MIN, MAX, COUNT, COUNTROWS, etc. for aggregating data.

- Logical Functions: Apply IF, SWITCH, AND, OR, NOT, etc. for conditional expressions and logic.
- Time Intelligence Functions: Use DATEADD, DATESYTD, SAMEPERIODLASTYEAR, etc. for time-based calculations.
- Filter Functions: Apply CALCULATE, FILTER, ALL, ALLEXCEPT, etc. to modify filter context and manipulate data.
- Iterator Functions: Use SUMX, AVERAGEX, MINX, MAXX, etc. to perform row-by-row calculations.
- Text Functions: Apply CONCATENATE, LEFT, RIGHT, MID, SUBSTITUTE, etc. for text manipulation.
- Parent-Child Functions: Use PATH, PATHLENGTH, PATHITEM, etc. for hierarchical data analysis.

DAX Evaluation Context

Understand the evaluation context for DAX calculations and how it affects results.

- Row Context: Represents the context in which each row of a table is evaluated.
- Filter Context: Represents the active filters applied to the data model during evaluation.
- Context Transition: Occurs when the row context is transformed into a filter context, typically within CALCULATE or CALCULATETABLE functions.

DAX Time Intelligence

Perform advanced time-based calculations using DAX time intelligence functions.

- Date Table: Create a dedicated date table in the data model to enable time-based calculations.
- Date Relationships: Establish relationships between the date table and fact tables in the data model.
- Time-Based Calculations: Calculate year-to-date, quarter-to-date, month-over-month, year-over-year, and other time-based metrics.
- Dynamic Date Ranges: Use DAX to create dynamic date ranges such as the last N days, rolling averages, or custom periods.

DAX Optimization

Optimize DAX expressions for better performance and faster data analysis.

- Simplify Expressions: Reduce complexity and simplify DAX expressions for faster evaluation.
- Use Variables: Store intermediate results in variables to avoid redundant calculations.
- Evaluate Context: Understand the evaluation context and avoid unnecessary context transitions.
- Function Selection: Choose the most efficient functions for specific calculations.

DAX Best Practices

Implement best practices for writing effective and maintainable DAX expressions.

- Consistent Formatting: Use consistent formatting and indentation to improve readability.
- Clear Naming: Assign meaningful names to calculated columns, measures, and variables.
- Commenting: Add comments to explain complex expressions or logic for future reference.
- Function Categorization: Organize DAX functions into categories for better management and maintenance.

DAX Patterns

Learn and apply common DAX patterns to solve real-world business problems.

- Ranking: Use RANKX, TOPN, and related functions to rank items based on a specific measure or criteria.
- Running Total: Calculate running totals or cumulative sums using CALCULATE and time intelligence functions.
- Moving Average: Compute moving averages using AVERAGEX and time intelligence functions.
- Pareto Analysis: Perform Pareto analysis using CALCULATE, RANKX, and other functions to identify the top contributors.

- Segmentation: Apply dynamic segmentation using CALCULATE and nested IF statements to categorize data into segments.
- ABC Analysis: Conduct ABC analysis using RANKX and SWITCH functions to classify items into different groups.
- Budget Allocation: Allocate budgets or targets across different periods or dimensions using CALCULATE and time intelligence functions.

DAX Debugging

Troubleshoot and debug DAX expressions to identify and fix issues.

- Error Messages: Understand common DAX error messages and their causes.
- Divide Function: Use the DIVIDE function to handle division errors and display alternate values when needed.
- Test Measures: Create test measures to isolate and validate specific parts of complex DAX expressions.
- Step-by-Step Debugging: Break down complex expressions into smaller components and test each component individually.

DAX with Power Query (M)

Combine DAX with Power Query (M) for advanced data transformation and analysis.

- Data Preparation: Use Power Query (M) for data extraction, transformation, and loading (ETL) before creating DAX calculations.
- Custom Columns: Create custom columns using Power Query (M) for static calculations or DAX for dynamic calculations.
- Data Model Optimization: Leverage both Power Query (M) and DAX to optimize the data model for performance and maintainability.

Power Query

Power Query Basics

Understand the fundamentals of Power Query for data extraction, transformation, and loading (ETL) in Power BI.

- Power Query Editor: Use the Power Query Editor to perform data transformations and create queries.
- Query Steps: Apply a series of steps to transform data, with each step recorded in the Applied Steps pane.
- M Formula Language: Leverage the M formula language to create custom functions and advanced data transformations.
- Data Types: Work with various data types such as number, text, date, time, datetime, duration, and binary.

Data Sources

Connect to a wide range of data sources using Power Query connectors.

- File Types: Access data from Excel, CSV, JSON, XML, and other file formats.
- Databases: Connect to SQL Server, Oracle, MySQL, PostgreSQL, and other databases.
- Cloud Services: Import data from Azure Blob Storage, Azure Data Lake, Google BigQuery, and more.
- Online Services: Access data from Salesforce, SharePoint, Dynamics 365, and other online services.
- Custom Connectors: Develop custom connectors to connect to specific data sources or APIs.

1. Data Extraction
   Extract data from various sources and formats using Power Query.

- Table Navigation: Navigate through tables, records, and lists to extract required data.
- Column Selection: Select specific columns to include or exclude from the dataset.
- Table Expansion: Expand nested tables, records, or lists to extract embedded data.
- File Combinations: Combine multiple files from a folder into a single dataset.

Data Transformation

Apply various data transformation functions to clean, reshape, and enrich the data.

- Column Operations: Rename, reorder, split, merge, and change the data type of columns.
- Row Operations: Filter, sort, remove duplicates, and split or group rows based on conditions.
- Text Transformations: Use trim, uppercase, lowercase, substring, and other text functions.
- Date and Time Transformations: Extract year, month, day, hour, minute, and second, and perform date arithmetic.
- Conditional Operations: Add conditional columns based on specific criteria or logic.
- Table Operations: Merge, append, and transpose tables, and create pivot or unpivot operations.
- Custom Functions: Create custom functions using M formula language for complex data transformations.

Data Loading

Configure data loading settings for Power Query in Power BI.

- Load to Model: Load transformed data directly into the Power BI data model.
- Load to Data Lake: Load data to an Azure Data Lake or other data storage for further processing.
- Incremental Load: Set up incremental load using Power Query to reduce data refresh time.
- DirectQuery: Connect to certain data sources using DirectQuery for real-time data access.

1. Data Shaping
   Shape data into the desired format and structure for reporting and analysis.

- Index Columns: Add index columns to maintain row order or create unique identifiers.
- Hierarchies: Create hierarchies to support drill-down functionality in Power BI visuals.

- Star Schema: Design data tables in a star schema with fact and dimension tables for efficient reporting and analysis.

Query Optimization

Optimize Power Query performance for faster data loading and processing.

- Remove Unnecessary Columns: Remove unused columns to reduce dataset size and improve performance.
- Load Summary Data: Load aggregated or summary data instead of detailed data when possible.
- Folding: Ensure query folding is applied to push transformation steps back to the data source.
- Cache Intermediate Results: Cache intermediate results to speed up query execution during development.

Power Query with DAX

Combine Power Query with DAX for advanced data transformation and analysis.

- Data Preparation: Use Power Query for data extraction, transformation, and loading (ETL) before creating DAX calculations.
- Custom Columns: Create custom columns using Power Query for static calculations or DAX for dynamic calculations.
- Data Model Optimization: Leverage both Power Query and DAX to optimize the data model for performance and maintainability.

Power Query Best Practices

Implement best practices for creating effective and maintainable Power Query transformations.

- Consistent Naming: Use consistent naming conventions for queries, columns, and functions.
- Query Organization: Organize queries into folders or groups for better management and maintenance.
- Commenting: Add comments to explain complex transformations or custom M code for future reference.

- Parameterization: Use parameters to make queries more flexible and adaptable to changing requirements.

M Programming Language

M Language Basics

Understand the fundamentals of M language for data transformation and manipulation in Power BI.

- Power Query: Utilize M language within Power Query Editor for advanced data transformation.
- Case Sensitivity: Remember that M language is case-sensitive, so be mindful of syntax and function names.
- Expressions: Use expressions to perform calculations, create new columns, or filter data.
- Steps: Apply a series of steps, each represented by an M expression, to transform data in Power Query.

Data Types

Work with various data types in M language, such as number, text, date, time, datetime, duration, and binary.

- Type Conversion: Convert data types using functions like Number.From, Text.From, DateTime.From, and others.
- Type Checking: Use functions like Value.Is, Type.Is, or Type.IsNullable to check data types.

Functions

Leverage built-in M functions for data transformation, extraction, and manipulation.

- Arithmetic Functions: Perform calculations using functions like Number.Add, Number.Multiply, and Number.Divide.
- Text Functions: Manipulate text with functions like Text.Upper, Text.Trim, Text.Replace, and Text.Length.
- DateTime Functions: Work with date and time using functions like DateTime.Add, DateTime.Day, and DateTime.LocalNow.

- Logical Functions: Apply logical conditions using functions like if, and, or, not, and List.Contains.
- List Functions: Manipulate lists with functions like List.First, List.Last, List.RemoveItems, and List.Transform.

Custom Functions

Create custom functions in M language to perform complex data transformations.

- Function Syntax: Define custom functions using the syntax (parameter as type) => expression.
- Recursive Functions: Build recursive functions for tasks like hierarchical data flattening or iterative calculations.

Error Handling

Handle errors and null values in M language to maintain data quality and integrity.

- Error Functions: Use functions like Error.Record, Error.Retry, and Error.Table to handle and manage errors.
- Null Handling: Manage null values with functions like Value.ReplaceType, Value.ReplaceNull, and Value.Remove.

Query Folding

Optimize performance by enabling query folding in M language to push transformation steps back to the data source.

- Supported Functions: Use functions that support query folding, such as Table.SelectColumns or Table.Sort.
- Custom Connectors: Develop custom connectors that support query folding for better performance.

Advanced Data Transformation

Apply advanced M language techniques for complex data transformation scenarios.

- Grouping and Aggregation: Use Table.Group to group data by specific columns and apply aggregations.
- Pivoting and Unpivoting: Reshape data using Table.Pivot and Table.Unpivot for more efficient analysis.

- Merging and Appending: Combine tables with Table.NestedJoin or Table.Combine for data consolidation.

M Programming Language Best Practices

Implement best practices for writing and maintaining M language code in Power BI.

- Consistent Naming: Use consistent naming conventions for functions, columns, and variables.
- Commenting: Add comments to M code to explain complex transformations or custom functions.
- Parameterization: Use parameters to make your M language code more flexible and adaptable.

Useful Links

- The Power BI Bootcamp: this Power BI Bootcamp will take you from absolute beginner in Power BI to being able to get hired as a confident and effective Business Intelligence Analyst. You'll learn the full Power BI suite of tools, the M programming language, and the DAX formula language, all while analyzing real-world datasets (including on fun topics like UFO and Bigfoot sightings!) and building enterprise-level projects.