[Power BI Dashboard Creation] [cheatsheet]

1. Data Import and Transformation (M)

- Import data from CSV: Source = Csv.Document(File.Contents("file.csv"), [Delimiter=",", Encoding=1252, QuoteStyle=QuoteStyle.None])
- Import data from Excel: Source = Excel.Workbook(File.Contents("file.xlsx"), null, true)
- Import data from SQL Server: Source = Sql.Database("server", "database", [Query="SELECT * FROM table"])
- Import data from web: Source = Web.Page(Web.Contents("https://example.com"))
- Filter rows: Table.SelectRows(Source, each [Column] > 10)
- Remove columns: Table.RemoveColumns(Source, {"Column1", "Column2"})
- Rename columns: Table.RenameColumns(Source, {{"OldName", "NewName"}})
- Change data type: Table.TransformColumnTypes(Source, {{"Column", type text}})
- Replace values: Table.ReplaceValue(Source, "OldValue", "NewValue", Replacer.ReplaceText, {"Column"})
- Merge queries: Table.NestedJoin(Source1, {"Key"}, Source2, {"ForeignKey"}, "NewColumn", JoinKind.LeftOuter)

2. Calculated Columns and Measures (DAX)

- Create a calculated column: Column = Table[Existing Column] * 2
- Create a measure: Measure = SUM(Table[Column])
- Calculate year-to-date (YTD) sales: YTD Sales = TOTALYTD(SUM(Sales[Amount]), 'Date'[Date])
- Calculate year-over-year (YoY) growth: YoY Growth = (SUM(Sales[Amount]) - CALCULATE(SUM(Sales[Amount]), SAMEPERIODLASTYEAR('Date'[Date]))) / CALCULATE(SUM(Sales[Amount]), SAMEPERIODLASTYEAR('Date'[Date]))
- Calculate running total: Running Total = CALCULATE(SUM(Sales[Amount]), FILTER(ALL(Sales), Sales[Date] <= MAX(Sales[Date])))
- Calculate percentage of total: % of Total = SUM(Sales[Amount]) / CALCULATE(SUM(Sales[Amount]), ALL(Sales))
- Calculate moving average: Moving Average = CALCULATE(AVERAGE(Sales[Amount]), DATESINPERIOD('Date'[Date], LASTDATE('Date'[Date]), -89, DAY))
- Rank values: Rank = RANKX(ALL(Sales), Sales[Amount])

- Categorize values: Category = IF(Sales[Amount] < 1000, "Low", IF(Sales[Amount] < 5000, "Medium", "High"))
- Calculate difference from previous period: Difference = Sales[Amount] - CALCULATE(SUM(Sales[Amount]), PREVIOUSMONTH('Date'[Date]))

3. Time Intelligence (DAX)

- Calculate total sales for current month: Total Sales Current Month = TOTALMTD(SUM(Sales[Amount]), 'Date'[Date])
- Calculate total sales for previous month: Total Sales Previous Month = CALCULATE(SUM(Sales[Amount]), PREVIOUSMONTH('Date'[Date]))
- Calculate total sales for current quarter: Total Sales Current Quarter = TOTALQTD(SUM(Sales[Amount]), 'Date'[Date])
- Calculate total sales for previous quarter: Total Sales Previous Quarter = CALCULATE(SUM(Sales[Amount]), PREVIOUSQUARTER('Date'[Date]))
- Calculate total sales for current year: Total Sales Current Year = TOTALYTD(SUM(Sales[Amount]), 'Date'[Date])
- Calculate total sales for previous year: Total Sales Previous Year = CALCULATE(SUM(Sales[Amount]), PREVIOUSYEAR('Date'[Date]))
- Calculate total sales for last 30 days: Total Sales Last 30 Days = TOTALMTD(SUM(Sales[Amount]), 'Date'[Date], -30)
- Calculate total sales for last 7 days: Total Sales Last 7 Days = CALCULATE(SUM(Sales[Amount]), DATESINPERIOD('Date'[Date], LASTDATE('Date'[Date]), -6, DAY))
- Calculate total sales for year-to-date (YTD) previous year: Total Sales YTD Previous Year = CALCULATE(TOTALYTD(SUM(Sales[Amount]), 'Date'[Date]), PREVIOUSYEAR('Date'[Date]))

4. Filtering and Slicing (DAX)

- Filter by single value: Filtered Sales = CALCULATE(SUM(Sales[Amount]), 'Product'[Category] = "Furniture")
- Filter by multiple values: Filtered Sales = CALCULATE(SUM(Sales[Amount]), 'Product'[Category] IN {"Furniture", "Office Supplies"})
- Filter by date range: Filtered Sales = CALCULATE(SUM(Sales[Amount]), 'Date'[Date] >= DATE(2022, 1, 1), 'Date'[Date] <= DATE(2022, 12, 31))
- Filter by top N values: Top 5 Products = TOPN(5, ALL('Product'[Product Name]), SUM(Sales[Amount]))
- Filter by bottom N values: Bottom 5 Products = BOTTOMN(5, ALL('Product'[Product Name]), SUM(Sales[Amount]))

- Exclude a specific value: Excluded Sales = CALCULATE(SUM(Sales[Amount]), 'Product'[Category] <> "Furniture")
- Filter by related table: Filtered Sales = CALCULATE(SUM(Sales[Amount]), FILTER(Customer, Customer[Country] = "USA"))
- Filter by calculated column: Filtered Sales = CALCULATE(SUM(Sales[Amount]), 'Product'[Price] > 100)
- Filter by measure: Filtered Sales = CALCULATE(SUM(Sales[Amount]), [Total Sales] > 1000000)

5. Aggregations and Grouping (DAX)

- Sum of values: Total Sales = SUM(Sales[Amount])
- Average of values: Average Sales = AVERAGE(Sales[Amount])
- Minimum value: Min Sales = MIN(Sales[Amount])
- Maximum value: Max Sales = MAX(Sales[Amount])
- Count of rows: Row Count = COUNT(Sales[Amount])
- Count of distinct values: Distinct Count = DISTINCTCOUNT(Sales[Customer])
- Group by and sum: Sales by Category = SUMMARIZE(Sales, 'Product'[Category], "Total Sales", SUM(Sales[Amount]))
- Group by and average: Average Sales by Category = SUMMARIZE(Sales, 'Product'[Category], "Average Sales", AVERAGE(Sales[Amount]))
- Group by and count: Count by Category = SUMMARIZE(Sales, 'Product'[Category], "Count", COUNT(Sales[Amount]))
- Group by multiple columns: Sales by Category and Country = SUMMARIZE(Sales, 'Product'[Category], 'Customer'[Country], "Total Sales", SUM(Sales[Amount]))

6. Conditional Formatting (DAX)

- Conditional background color: Background Color = IF([Total Sales] > 1000000, "Green", "Red")
- Conditional font color: Font Color = IF([Total Sales] > 1000000, "White", "Black")
- Conditional icon: Icon = IF([Total Sales] > 1000000, "Smiley", "Frowny")
- Conditional data bars: Data Bars = IF([Total Sales] > 1000000, 1, 0)
- Conditional KPI: KPI = IF([Total Sales] > [Target Sales], "Above Target", "Below Target")

7. Table and Matrix Visuals (DAX)

- Conditional row color: `Row Color = IF([Total Sales] > 1000000, "Green", "Red")`
- Conditional column color: `Column Color = IF([Total Sales] > 1000000, "Green", "Red")`
- Conditional cell color: `Cell Color = IF([Total Sales] > 1000000, "Green", "Red")`
- Conditional formatting based on another column: `Color = IF([Category] = "Furniture", "Blue", "Gray")`
- Conditional formatting based on a measure: `Color = IF([Total Sales] > [Average Sales], "Green", "Red")`
- Data bars in a table: `Data Bars = IF(NOT(ISBLANK([Total Sales])), [Total Sales], BLANK())`
- KPIs in a table: `KPI = IF([Total Sales] > [Target Sales], "Above Target", "Below Target")`

8. Custom Visuals (HTML, CSS, JavaScript)

- Create a custom visual: Tutorial: Develop a Power BI circle card visual
- Package a custom visual: `pbiviz package`
- Install a custom visual: `pbiviz install`
- Update a custom visual: `pbiviz update`

9. Tooltips and Report Interactions (DAX)

- Custom tooltip: `Tooltip = "Sales: " & FORMAT([Total Sales], "C") & " | Profit: " & FORMAT([Total Profit], "C")`
- Conditional tooltip: `Tooltip = IF([Total Sales] > 1000000, "High Sales", "Low Sales")`
- Drillthrough to another page: `Drillthrough = SELECTEDVALUE('Product'[Category])`
- Drillthrough with multiple fields: `Drillthrough = SELECTEDVALUE('Product'[Category]) & " - " & SELECTEDVALUE('Date'[Year])`
- Conditional drillthrough: `Drillthrough = IF([Total Sales] > 1000000, SELECTEDVALUE('Product'[Category]), BLANK())`
- Filter other visuals on click: `Visual Interaction = SELECTEDVALUE('Product'[Category])`

10. Bookmarks and Navigation (DAX)

- Create a bookmark: `Bookmarks > Add`
- Link to a bookmark: `Action > Navigate to Bookmark`

- Conditional bookmark: Bookmark Visibility = IF([Total Sales] > 1000000, TRUE, FALSE)
- Toggle bookmark: Bookmark Interaction = SELECTEDVALUE('Bookmark'[Bookmark Name])
- Navigation with bookmarks: Navigation > Bookmarks

11. Themes and Formatting (JSON)

- Import a custom theme: View > Themes > Browse for Themes
- Customize theme colors: "dataColors": ["#FF0000", "#00FF00", "#0000FF"]
- Customize theme fonts: "textClasses": [{ "fontSize": 12, "fontFamily": "Arial" }]
- Customize theme visualStyles: "visualStyles": {"*": {"*": {"border": true}}}
- Conditional formatting with themes: "visualStyles": {"matrix": {"cell": {"*": {"background": {"solid": {"color": {"expr": {"if": [{">=": [{"field": "Sales"}, 1000]}, "#00FF00", "#FF0000"]}}}}}}}}

12. Report Layout and Design (JSON)

- Set page size: "pageSize": {"type": "A4", "orientation": "landscape"}
- Set page background: "pageBackground": {"color": "#FFFFFF"}
- Set visual spacing: "visualSpacing": 10
- Set visual alignment: "visualAlignment": "left"
- Set visual borders: "visualBorders": true
- Set visual header: "visualHeader": true
- Set visual background: "visualBackground": {"color": "#FFFFFF"}
- Set visual title: "title": {"text": "Sales Report"}
- Set visual subtitle: "subtitle": {"text": "Fiscal Year 2022"}
- Set visual legend: "legend": {"position": "Right"}

13. Data Security and Row-Level Security (DAX)

- Define roles: Manage Roles > Create New Role
- Define role filters: 'Product'[Category] = "Furniture"
- Assign roles to users: Manage Roles > Assign Users
- Dynamic row-level security: Security Filter = USERNAME()
- Conditional row-level security: Security Filter = IF(USERNAME() = "Admin", ALL('Product'[Category]), 'Product'[Category] = "Furniture")

14. Query Parameters and Dynamic Filtering (M)

- Define a parameter: `Category = "Furniture"`
- Use a parameter in a query: `Table.SelectRows(Source, each [Category] = Category)`
- Define a function with parameters: `FilterByCategory = (table as table, category as text) => Table.SelectRows(table, each [Category] = category)`
- Invoke a function with parameters: `FilteredTable = FilterByCategory(Source, "Furniture")`
- Dynamic filtering with parameters: `FilteredTable = Table.SelectRows(Source, each [Category] = Parameter("SelectedCategory"))`

15. Advanced Analytics and AI (DAX, R, Python)

- Time series forecasting: `Forecast = FORECAST.ETS(Sales[Amount], 'Date'[Date])`
- Linear regression: `Regression = LINEST(Sales[Amount], 'Date'[Date])`
- Clustering: `Cluster = CLUSTERSET(Customer[Age], Customer[Income], 3)`
- Classification: `Class = CLASSIFICATIONSET(Customer[Age], Customer[Income], Customer[Segment])`
- R script execution: `R Script = RSCRIPT("library(forecast); forecast(ts(Sales[Amount]), h=12)")`
- Python script execution: `Python Script = PYTHON("import pandas as pd; df = pd.DataFrame(Sales); df.head()")`