Build a DRIP Modeling Dashboard

Part II : Data Visualization ~ 30 July 2025

Pre-Requisites:

- Download Power BI Desktop (totally free).
  - If you are a MAC user, here is the quickest solution (use the free trial).
- Download

Custom Background:

Use a custom background (PNG created in PowerPoint) to create sleek containers for your visuals.

- Download the Drip Modeling Background.png
-

Requirement: Create a Dividend Reinvestment Plan (DRIP) *Modeling* Dashboard using **forecasting-focused data** with built-in **simulation parameters**. This dashboard should use only free (but reputable) data sources. The dashboard should provide a means to test different stocks, dividend yields, reinvestment assumptions, and time horizons.

Table	Description
CompanyTickers	Complete master list of all company stock tickers
All Stocks Overview	Tickers to test in DRIP simulation with metadata and inputs
DividendProjections	Measures for earnings, reinvestments, share growth
DateTable	Calendar table for time-based visuals
Parameters	What-If Parameters for user input: years, contributions, growth rate

Visualizations:

- 📈 ***"Projected Dividend Income Over Time"***
- 🔁 ***"Shares Owned with DRIP vs. No DRIP"***
- 💰 ***"Annual Reinvested Dividends and Growth"***
- 🧮 ***"Yield on Cost by Year"***
- 🎯 ***"Compare Multiple Stocks for Income Potential"***

Bonus Features:

- Add multiple tickers and weight them
- Allocate different $ amounts per stock

- Compare income curves side-by-side

Let's Get Started!!

Pre-Requisites:

- Download Power BI Desktop (totally free).
- Get API key from Free Stock APIs in JSON & Excel | Alpha Vantage

Data Sources:

In Power BI, go to Home tab > New Source > More > Search for the connector (e.g., JSON) > Connect.

1. **CompanyTickers:**
    - **Connector:** JSON
    - **Source:** SEC.gov | Company Tickers → copy/past content and save with .json file extension
    - **Type:** Manual (export/import)
    - **Role in Data Model:** Fact table (center of the star schema)
    - **Query Name:** Under Queries (on left) be sure to rename the table CompanyTickers (NO SPACES)

2. **AllStocksOverview:** We'll build this table in three steps:
    - **TickerList:**
        - Use 'Enter Data' and manually input selected tickers in a single column
          Start with these: CVX, PEP, HRL, ESS, ED
        - **Column Name:** Be sure to name the column Ticker
        - **Query Name:** Be sure to rename this table TickerList (NO SPACES)

    - **GetStockOverview:**
        - In Power Query Editor, Home tab > New Source > **Blank Query**
        - Click Advanced Editor (center of Home tab).
        - Copy/paste the following M code to create a custom function:
            - Be sure to copy all the way to 'Output' (on next page)
            - Replace ENTER_API_KEY with your AlphaVantage API key (see pre-requisites).
        - **Query Name:** Be sure to rename this function GetStockOverview

```
(ticker as text) =>
let
  apiKey = "ENTER_API_KEY",

  // Fetch company overview
  overviewResponse = Json.Document(
    Web.Contents("https://www.alphavantage.co", [
      RelativePath = "query",
      Query = [
        function = "OVERVIEW",
        symbol = ticker,
        apikey = apiKey
      ]
    ])
  ),

  // Fetch current price from GLOBAL_QUOTE
  quoteResponse = Json.Document(
    Web.Contents("https://www.alphavantage.co", [
      RelativePath = "query",
      Query = [
        function = "GLOBAL_QUOTE",
        symbol = ticker,
        apikey = apiKey
      ]
    ])
  ),

  // Parse quote record safely
  quoteRecord = try Record.Field(quoteResponse, "Global Quote") otherwise [],

  // Extract latest price
  latestPrice = try Number.FromText(Record.FieldOrDefault(quoteRecord, "05. price", "0")) otherwise null,

  // Final output with all fields
  Output = [
    Ticker = ticker,
    CompanyName = Record.FieldOrDefault(overviewResponse, "Name", null),
    Sector = Record.FieldOrDefault(overviewResponse, "Sector", null),
    Industry = Record.FieldOrDefault(overviewResponse, "Industry", null),
    MarketCapBillion = try Number.FromText(Record.FieldOrDefault(overviewResponse, "MarketCapitalization", "0")) / 1000000000
otherwise null,
    CurrentPrice = latestPrice,
    DividendPerShare = try Number.FromText(Record.FieldOrDefault(overviewResponse, "DividendPerShare", "0")) otherwise 0,
    DividendGrowthRate = try Number.FromText(Record.FieldOrDefault(overviewResponse, "FiveYearAvgDividendYield", "0")) / 100
otherwise 0,
    PayoutRatio = try Number.FromText(Record.FieldOrDefault(overviewResponse, "PayoutRatio", "0")) otherwise null
  ]
in
  Output
```

- **AllStocksOverview:**
    - In Power Query Editor, Home tab > New Source > **Blank Query**
    - Click Advanced Editor (center of Home tab).
    - Copy/paste the following M code to combine test stocks with AlphaVantage data
    - **Query Name:** Be sure to rename the table AllStocksOverview

```
let
  Tickers = TickerList,
  Expanded = Table.AddColumn(Tickers, "OverviewData", each
GetStockOverview([Ticker])),
  #"Expanded OverviewData" = Table.ExpandRecordColumn(Expanded, "OverviewData",
{"Ticker", "CompanyName", "Sector", "Industry", "MarketCapBillion", "CurrentPrice",
"DividendPerShare", "DividendGrowthRate", "PayoutRatio"}, {"OverviewData.Ticker",
"OverviewData.CompanyName", "OverviewData.Sector", "OverviewData.Industry",
"OverviewData.MarketCapBillion", "OverviewData.CurrentPrice",
"OverviewData.DividendPerShare", "OverviewData.DividendGrowthRate",
"OverviewData.PayoutRatio"})
in
  #"Expanded OverviewData"
```

★ **IMPORTANT**: AlphaVantage only allows 25 free API calls per day, so be careful not to click 'Refresh' too often in a day.

3. **2025DividendArtistocrats:**
    - **Connector:** CSV
    - **Source:** 2025 Dividend Aristocrats List: All 69 Ranked & Analyzed → scroll down to table and 'Export' the data to a CSV file.
    - **Type:** Manual (export/import)
    - **Role in Data Model:** Bonus info about Dividend Aristocrats
    - **Query Name:** Under Queries (on left) be sure to rename the table 2025DividendAristocrats (NO SPACES)

4. **2025MonthlyDividendStocks:**
    - **Connector:** CSV
    - **Source:** 2025 Monthly Dividend Stocks List: All 76 Ranked and Analyzed → scroll down to table and 'Export' the data to a CSV file.
    - **Type:** Manual (export/import)
    - **Role in Data Model:** Bonus info about all dividend stocks (includes 'Dividend Safety')
    - **Query Name:** Under Queries (on left) be sure to rename the table 2025MonthlyDividendStocks (NO SPACES)

You should now have 6 queries in Power Query Editor:

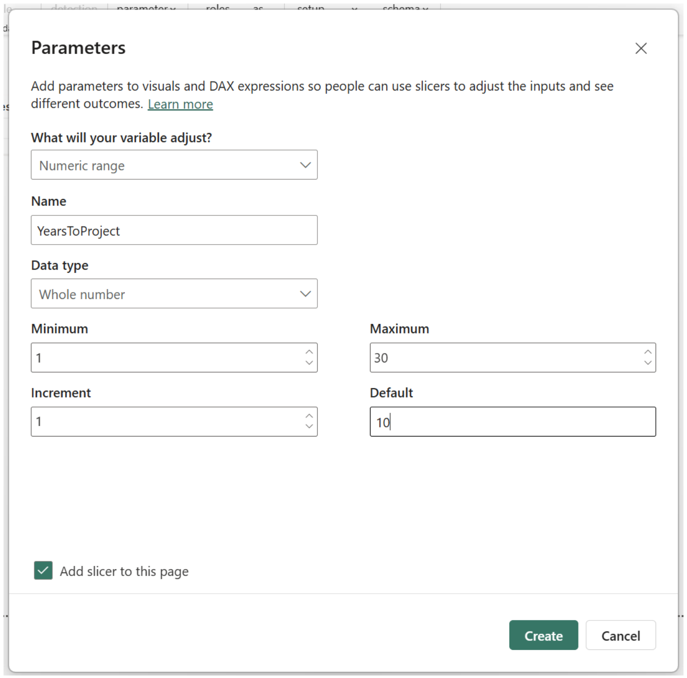
- **CompanyTickers:** A master table of all stock tickers.
- **TickerList:** Your manually-entered stocks you want to test.
- **GetStockOverview:** A function that calls the AlphaVantage's OVERVIEW API.
- **AllStocksOverview:** A custom table that combines your TickerList with the API data.
- **2025DividendArtistocrats:** A table of current Dividend Aristocrats.
- **2025MonthlyDividendStocks:** A table of stocks that provide monthly dividends.

Once you are done loading your data connections in Power Query Editor, click **Close & Apply** (top-left) to load the data into the model in Power BI.

What-If Parameters:

In **Report View**, go to Modeling tab, New Parameter > **Numeric Range** and create the following four 'What-If' parameters for use in projecting DRIP investment strategies.

1. **YearsToProject**



Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. Learn more

What will your variable adjust?
Numeric range

Name
YearsToProject

Data type
Whole number

Minimum
1

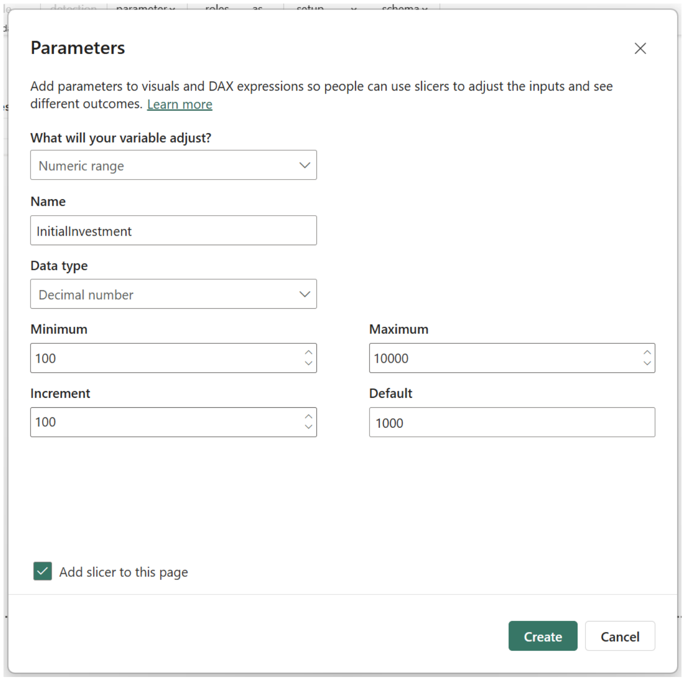
Maximum
30

Increment
1

Default
10

☑ Add slicer to this page

2. **InitialInvestment**



Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. Learn more

What will your variable adjust?
Numeric range

Name
InitialInvestment

Data type
Decimal number

Minimum
100

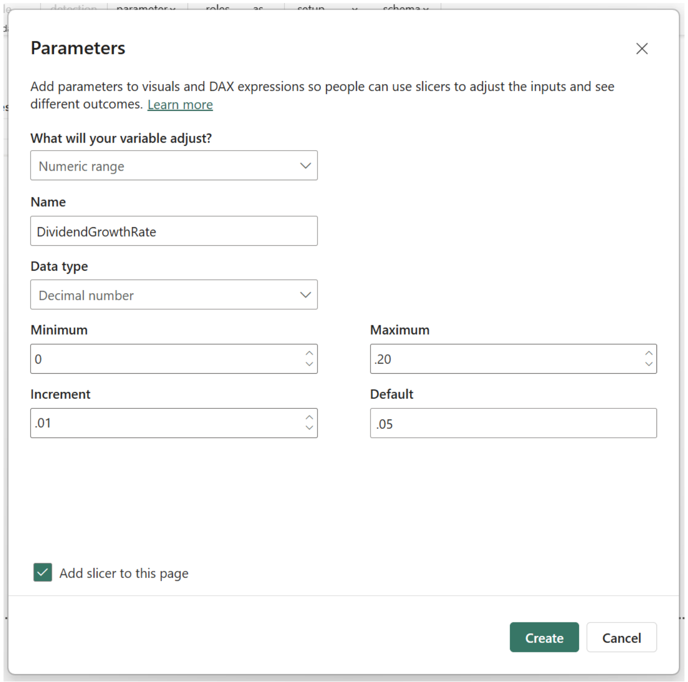
Maximum
10000

Increment
100

Default
1000

☑ Add slicer to this page

3. **DividendGrowthRate**



Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. Learn more

What will your variable adjust?
Numeric range

Name
DividendGrowthRate

Data type
Decimal number

Minimum
0

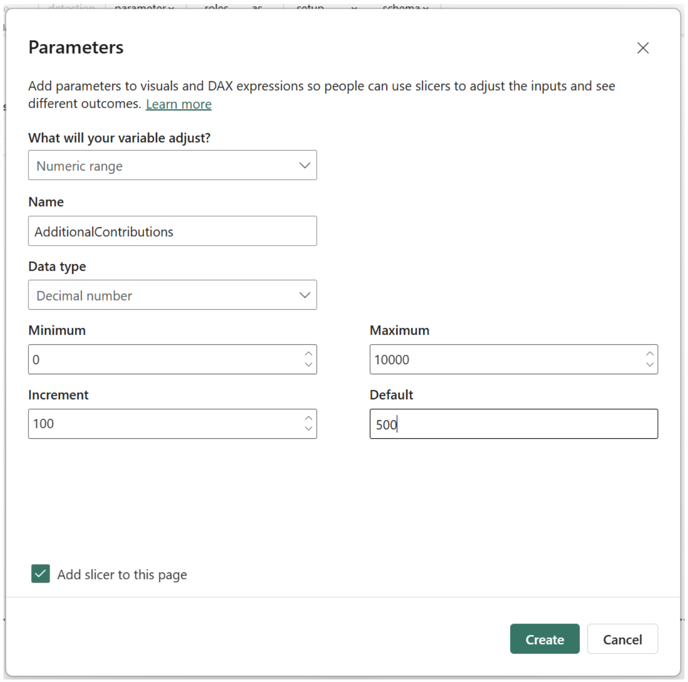
Maximum
.20

Increment
.01

Default
.05

☑ Add slicer to this page

4. **AdditionalContributions**



Parameters

Add parameters to visuals and DAX expressions so people can use slicers to adjust the inputs and see different outcomes. Learn more

What will your variable adjust?
Numeric range

Name
AdditionalContributions

Data type
Decimal number

Minimum
0

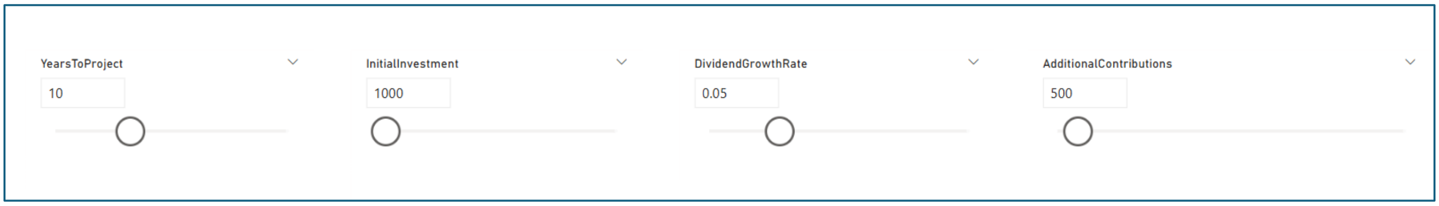
Maximum
10000

Increment
100

Default
500

☑ Add slicer to this page

You should now see four slicers on your report – one for each What-If parameter:



YearsToProject
10

InitialInvestment
1000

DividendGrowthRate
0.05

AdditionalContributions
500

Projections Table:

In **Report View**, go to Modeling tab > **New Table** then copy/paste the following DAX to create a *calculated table* called DividendProjections.

(NOTE: Be sure to copy the entire code, which continues on the next page.)

```
DividendProjections =
VAR StartYear = YEAR(TODAY())
VAR EndYear = StartYear + [YearsToProject Value] - 1
VAR Investment = [InitialInvestment Value]
VAR GrowthRateOverride = [DividendGrowthRate Value]
VAR MonthlyContribution = [AdditionalContributions Value]

RETURN
GENERATE (
    AllStocksOverview,
    ADDCOLUMNS (
        CALENDAR (DATE(StartYear, 1, 1), DATE(EndYear, 12, 31)),

        // Time breakdown
        "Year", YEAR([Date]),
        "Month", MONTH([Date]),
        "YearMonth", FORMAT([Date], "YYYY-MM"),

        // Starting shares based on initial investment
        "StartingShares", Investment / [OverviewData.CurrentPrice],

        // Total investment including recurring contributions
        "TotalContributions",
            Investment +
            DATEDIFF(DATE(StartYear, 1, 1), [Date], MONTH) * MonthlyContribution,

        // Total shares simulated (flat pricing assumption)
        "TotalShares",
            (
                Investment +
                DATEDIFF(DATE(StartYear, 1, 1), [Date], MONTH) * MonthlyContribution
            ) / [OverviewData.CurrentPrice],

        // Projected DPS (with override applied)
        "DividendPerShareProjected",
            [OverviewData.DividendPerShare] *
            POWER(1 + GrowthRateOverride, YEAR([Date]) - StartYear),

        // Simulated dividend income for period
        "DividendIncome",
            (
                [OverviewData.DividendPerShare] *
                POWER(1 + GrowthRateOverride, YEAR([Date]) - StartYear)
            ) * (
                (
                    Investment +
                    DATEDIFF(DATE(StartYear, 1, 1), [Date], MONTH) * MonthlyContribution
                ) / [OverviewData.CurrentPrice]
            )
    )
)
```

Date Table:

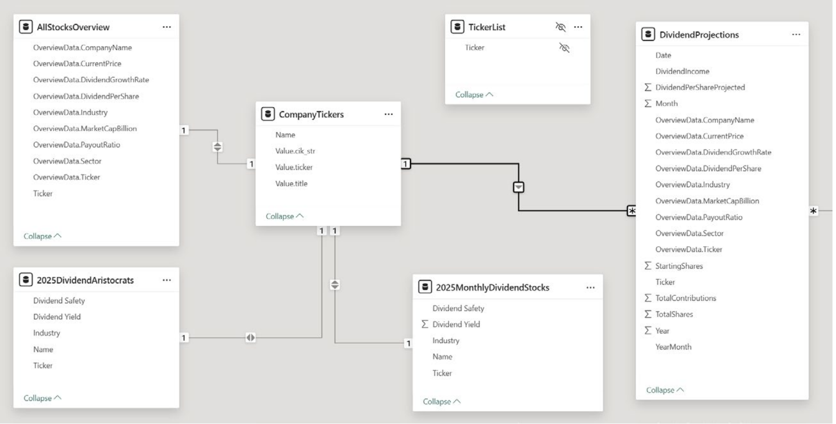
In **Report View**, go to Modeling tab > **New Table** then copy/paste the following DAX to create a DateTable which should always be used with time-series data.

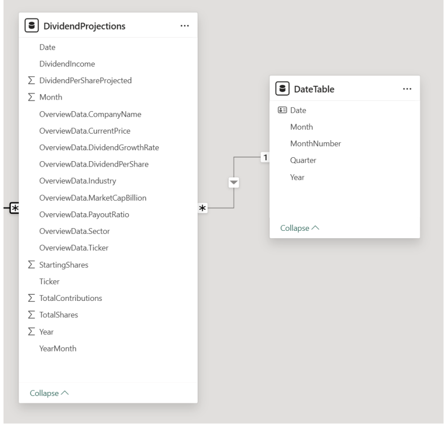
```
DateTable =
ADDCOLUMNS (
    CALENDAR (DATE(2025, 1, 1), DATE(2045, 12, 31)),  -- Adjust years as needed
    "Year", YEAR([Date]),
    "Month", FORMAT([Date], "MMM"),
    "MonthNumber", MONTH([Date]),
    "Quarter", "Q" & FORMAT([Date], "Q")
)
```

Data Model:

In **Model View**, create the relationships between your tables, making a 'star schema'.

- **Ticker:** Relate the Ticker the AllStocksOverview, DividendProjections, 2025MonthlyArtistocrats, 2025MonthlyDividendStocks to the Value.ticker field in CompanyTickers (the center of the star).
- **Date:** Relate the Date field on the DividendProjections and DateTable.



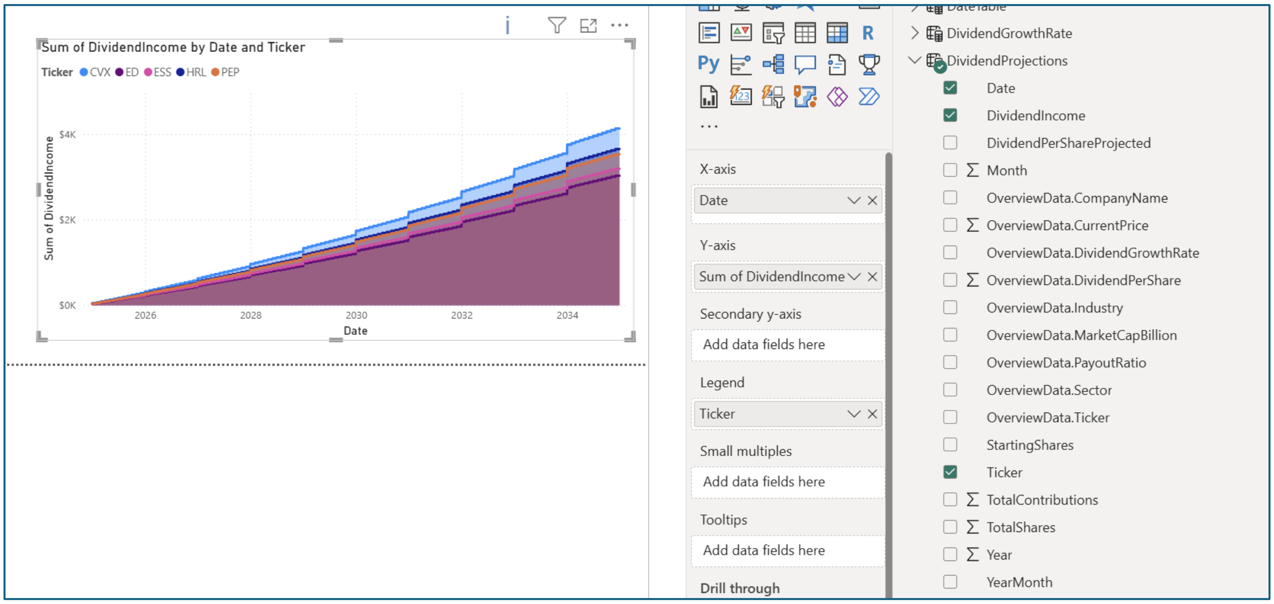


First Visuals:

Back in **Report View,** start adding visuals to our dashboard.

- **Projected Dividend Income Over Time:**
    - From the Visualizations pane, select the **Area Chart**
    - From the Data pane > DividendProjections table…
        - Drag the **Date** field to the X-axis
          IMPORTANT: Ensure date hierarchy is disabled.
        - Drag the **DividendIncome** field to the Y-axis
            - Switch 'Count' to 'Sum'

- Drag the **Ticker** field to Legend



X-axis

Date

Y-axis

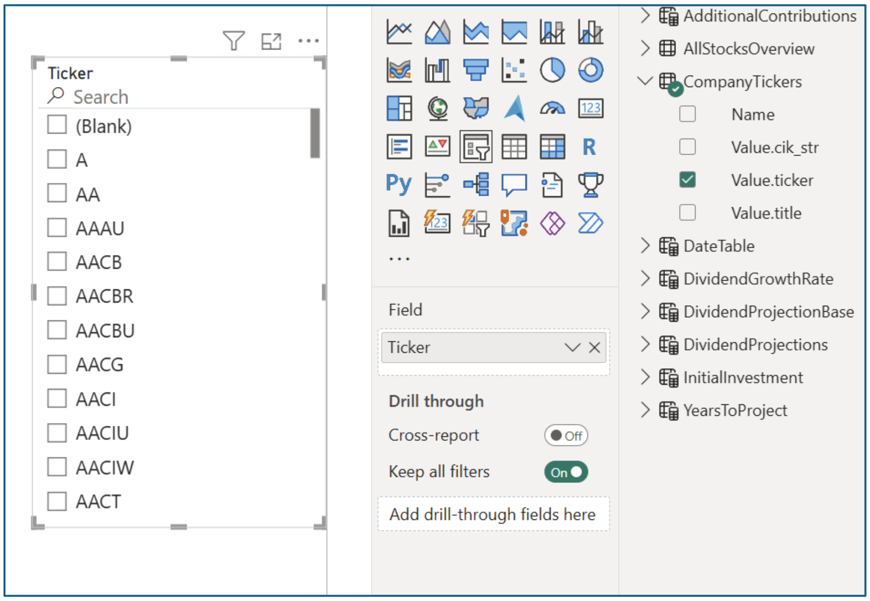
Sum of DividendIncome

Secondary y-axis

Legend

Ticker

- **Ticker:**
  - From the Visualizations pane, select the **Slicer**
  - From the Data pane > CompanyTickers table…
    - Drag the **Value.ticker** field to the visual
    - On the visualization, select the field > Rename for this visual and name it **Ticker**



Field

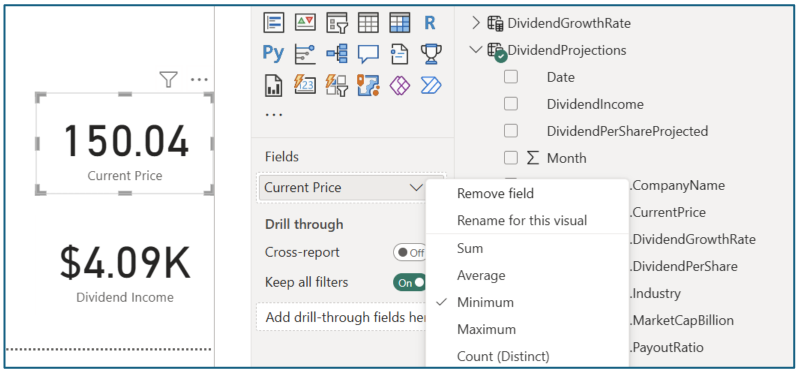
Ticker

Cross-report ○ Off

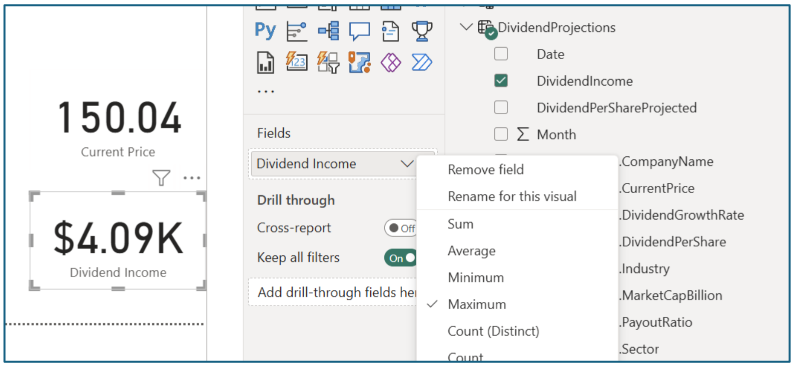
Keep all filters On ●

- **Current Price:**
  - From the Visualizations pane, select the **Card**
  - From the Data pane > DividendProjections table…
    - Drag the **OverviewData.CurrentPrice** to visual
    - Set visual to **Minimum**

- Rename in visual **Current Price**



- **Dividend Income:**
  - From the Visualizations pane, select the **Card**
  - From the Data pane > DividendProjections table…
    - Drag the **DividendIncome** to visual
    - Set visual to **Maximum**
    - Rename in visual **Dividend Income**



…That's it for Part I of this workshop. In Part II, we'll **complete** our DRIP Modeling Dashboard with more visuals and formatting (to make it pretty.)

What We Learned:

This work is sometimes called ETF (Extract, Transform, Load). In Power BI, we learned how to do…

Extract

Connectors:

- **Web –** We used the web connector to get data from AlphaVantage's API
- **JSON –** We used the JSON connector to get the ticker list from the SEC site.

- **CSV –** We used the CSV connector to load data from SimplySafeDividends site.

Manual:

- **Enter Data –** We manually entered the Tickers we want to analyze.

Transform (Power Query Editor)

- **Functions –** We created a function to connect to AlphaVantage's OVERVIEW API.
- **Queries –** We renamed the queries and provided a description (for our future selves).

Load (Data Model)

- **Relationships –** We used the Ticker and Date fields to create relationships between our tables.
- **Layouts –** We created a new layouts to differentiate between relationship types.

Calculations

- **Date Table –** We created a Date Table for normalizing time-based data.
- **Calculated Tables –** We created a DividendProjections table with calculations.

Parameters

- **What-If –** We created four 'What-If' parameters for testing different DRIP scenarios.

Visualizations

- **Slicer –** We created a 'Ticker' slicer for fast filtering of stocks.
- **Area Chart –** We created an area chart for comparing multiple stocks' dividend growth.
- **Card –** We used (single-row) cards to show both current price vs. projected dividend earnings.