# Synthetic Data Creation

## Expectations

I wanted to create data bases on a statistical method. I decided to make the Units Sold and the Price Sold variable with the inflation rate. As such, I do know that Inflation Rate should affect the Units Sold either positively or negatively. If I had left it that way, there should be an R Squared of 1 on the data for the prediction on Inflation Rate/Store ID.

I decided to force a difference based on a little bit of randomisation. The Units Sold were later randomised slightly by -3 to 3, to imitate a little bit of free will.

## Method

This data set evolved over 2 days.

I started by typing out my requirements into a single sheet and pasting it in. The problems there were that the Analysis crashed most of the time, or that when it did succeed, the dataset output did not include all of the specifications.

Eventually, I took the approach of building the data set up statement by statement. This was quicker to analyse and rarely failed. I exported the values in csv at every stage in case the connection reset. That made it possible to upload the data set and continue from that point.

Chat-GPT4 kept changing my numbers. There were times when I thought it was resolved, but Chat-GPT recalculated my Price Sold data, making prices into the 10s of thousands.

## Data Creation via Chat-GPT 4

I need to create some data. I need 8 unique Product IDs for each of these Product Categories ['Mens', 'Womens']. Product IDs must be numeric. Products 001-008 would map to Mens and 009-018 would be Womens. I need 3 unique Store IDs for each of these Countries ['UK', 'Ireland', 'France']. The store ID should be numeric. So the Store IDs 1- 3 would be Ireland, 4-6 would be UK, 7-9 France, etc... I would also like to include Inflation Rates by country by month and GDP Growth Rate calculations by country by month. These can be set with random numbers. Inflation Rate and GDP Growth Rate should be consistent for each country and month combination and be displayed on each row. We can use randomly generated data for inflation with values between -1 and 11 rounded to 2 decimal places. GDP Growth Rate can be randomised between -2 and 4 and should be consistent for each Month and Country. It should be rounded to 2 decimal points. We'll need Monthly Data from 2023-01 to 2024-04

ChatGPT

**ChatGPT**

I'll create a dataset that includes the following:

1. **Product IDs**: 16 unique numeric IDs, with IDs 001-008 assigned to "Mens" and 009-016 assigned to "Womens".
2. **Store IDs**: 9 unique numeric IDs, with IDs 1-3 assigned to Ireland, 4-6 to the UK, and 7-9 to France.
3. **Inflation Rates and GDP Growth Rates**: Random values for each country by month, from January 2023 to April 2024. Each combination of country and month will have consistent inflation and GDP growth rates.

I'll then compile this data into a comprehensive dataset. Let's start with the random generation of the economic data for each month and country, and then we'll put everything together.

I've generated the economic data for inflation rates and GDP growth rates for each month from January 2023 to April 2024 for Ireland, the UK, and France. Now, I'll create the full dataset that includes:

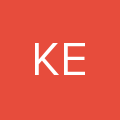
* Product IDs linked to categories ("Mens" and "Womens")
* Store IDs linked to countries (Ireland, UK, France)
* Monthly inflation rates and GDP growth rates for each country

Let's put it all together into a comprehensive dataset. ​​

The dataset is now complete. It includes:

* **Product IDs** (1-8 for Mens, 9-16 for Womens) linked to product categories.
* **Store IDs** (1-3 for Ireland, 4-6 for UK, 7-9 for France) linked to each country.
* **Monthly economic data** from January 2023 to April 2024, featuring consistent inflation rates and GDP growth rates for each country and month.

This data is comprehensive, including multiple entries per month for each combination of product and store ID, across all specified countries. If you need this data in a specific format (e.g., CSV, Excel) or have any further adjustments, please let me know! ​​



**You**

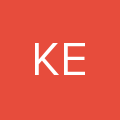
download scv please

ChatGPT

**ChatGPT**

The dataset has been saved as a CSV file. You can download it using the link below:

Download Product\_Store\_Economic\_Data.csv ​​



**You**

add a datepoint for every date in the month and name that 'Date of Sale'

ChatGPT

**ChatGPT**

To add a "Date of Sale" column that includes a unique date for each day in the month corresponding to each entry, I'll expand the dataset. For each existing row, which currently corresponds to the first day of the month, I'll create additional rows for each day in that month. Let's proceed with this modification.

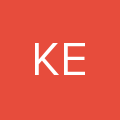
Error analyzing

It seems there was an error in the process. I'll correct this and ensure that the correct number of days in the month is used for each entry. Let's fix this and generate the expanded dataset again. ​​

The dataset has been successfully expanded to include a "Date of Sale" for every day of each month. Now, I'll save this expanded dataset to a CSV file for you to download. Let's proceed with that. ​​

The expanded dataset, now including a "Date of Sale" for every day in each month, has been saved as a CSV file. You can download it using the link below:

Download Expanded Product\_Store\_Economic Data.csv ​​



**You**

add a column for ['Price Sold']. Start this as a random number between 20 and 40 on the first date. Round it to 2 decimal places. At every change in the month, set the ['Price Sold'] to be (['Price Sold'] \* [Inflation Rate]) + [Price Sold]

ChatGPT

**ChatGPT**

To implement the requirement for the 'Price Sold' column:

1. I'll start by generating a random initial price between 20 and 40 for each product in each store on the first day of each month, rounding it to two decimal places.
2. On subsequent days of the month, the 'Price Sold' will be adjusted daily based on the formula (Price Sold×Inflation Rate/100)+Price Sold(Price Sold×Inflation Rate/100)+Price Sold, where the Inflation Rate is used as a percentage.

Let's proceed to update the dataset with these specifications.

Error analyzing

It seems I encountered an error in setting the 'Price Sold' values due to a mismatch in the lengths of the data. I'll need to adjust the approach to correctly apply the inflation rate calculations across each month and ensure each price adjusts daily. Let me correct this and generate the updated prices. ​​

Error analyzing

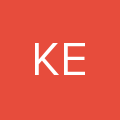
It looks like my method for recalculating the prices still has an issue with indexing after grouping. Let me adjust the approach to correctly apply the inflation adjustments and ensure all prices are updated correctly for each day in the dataset. I'll resolve this and generate the correct 'Price Sold' values. ​​

Error analyzing

I encountered another error due to an issue with setting values in the DataFrame. I will simplify the approach and ensure that the 'Price Sold' is calculated accurately for each day, without encountering indexing issues. Let me correct this and provide you with the updated dataset. ​​

The dataset has now been updated to include the 'Price Sold' column. Prices start with a random value between 20 and 40 on the first day of each month and adjust daily based on the inflation rate for each respective country and date.

If everything looks good, I can proceed to save this updated dataset as a CSV file for you to download. Let me know if you'd like me to proceed or make any further adjustments! ​​



**You**

instead of changing the 'price sold' for each day of the month, please do the calculation only on the changes of the 'Date' field, not the 'Date of Sale'

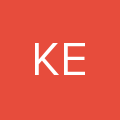
ChatGPT

**ChatGPT**

Understood! I will adjust the 'Price Sold' calculation to only update on the change of the 'Date' field at the start of each new month, rather than daily. This means the price will remain constant throughout a month and will only update at the beginning of each new month based on the inflation rate. Let's update the dataset accordingly.

The dataset now updates the 'Price Sold' only at the start of each new month, maintaining the same price throughout the month for each product and store. This adjustment ensures that prices change monthly based on the initial 'Date' field, not daily with 'Date of Sale'.

If this meets your requirements, I can proceed to save the updated dataset as a CSV file for you to download. Let me know if you'd like to proceed or if there are any other adjustments you need! ​​



**You**

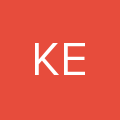
download csv

ChatGPT

**ChatGPT**

The updated dataset has been saved as a CSV file. You can download it using the link below:

Download Final Product\_Store\_Economic Data.csv ​​



**You**

Now we need to add a ['Units Sold'] column. This will start as a random integer between 10 and 30. At every change in the month, set the ['Units Sold] to be (['Units Sold'] \* [Inflation Rate]) + [Units Sold]

ChatGPT

**ChatGPT**

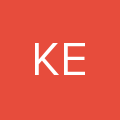
To add the 'Units Sold' column:

1. **Initial 'Units Sold'**: This will start as a random integer between 10 and 30 for each product in each store on the first day of each month.
2. **Monthly Adjustment**: At the start of each new month, 'Units Sold' will be recalculated based on the formula (Units Sold×Inflation Rate/100)+Units Sold(Units Sold×Inflation Rate/100)+Units Sold, where the Inflation Rate is used as a percentage.

Let's proceed with these updates to the dataset.

The 'Units Sold' column has been successfully added to the dataset, starting with random initial values between 10 and 30, and updating monthly based on the inflation rate for each product in each store. This update occurs at the beginning of each new month as specified.

If you're ready, I can proceed to save this final updated dataset as a CSV file for you to download. Let me know if you'd like to proceed or if there's anything else you need! ​​



**You**

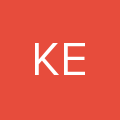
download csv

ChatGPT

**ChatGPT**

The final dataset, now including the 'Units Sold' column with monthly adjustments, has been saved as a CSV file. You can download it using the link below:

Download Final Product\_Store\_Economic Data with Units Sold.csv ​​



**You**

Let's add some changes to the units sold to make it a little more randomised. At each row, make the Units Sold = Units Sold + (a random integer between -3 and 3 for each row), but no lower than 0.

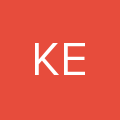
ChatGPT

**ChatGPT**

To make the 'Units Sold' values more varied day-to-day within each month, I'll adjust the dataset by adding a random integer between -3 and 3 to the 'Units Sold' for each row. The result will always be clamped to a minimum of 0 to ensure we don't have negative sales. Let's apply these changes.

The 'Units Sold' values have been updated to include random day-to-day variability within each month, with adjustments ranging between -3 and +3, ensuring no negative sales.

I can now save this adjusted dataset to a CSV file for you to download. Let me know if you'd like to proceed with this or if there are any other modifications you need! ​​



**You**

download csv

ChatGPT

**ChatGPT**

The dataset, now including the randomized 'Units Sold' adjustments, has been saved as a CSV file. You can download it using the link below:

Download Randomized Units Sold Product Store Economic Data.csv ​​

Top of Form

After testing, I found that the Price Sold seems to have been randomised, or at least recalculated at a store level. I want the price sold to match across the country.

User

**You**

Complete\_Sales\_Data.csv

Spreadsheet

I need each store for the country for the month to have the same price for each product. For example, if Store 1 has a price of 28.99 for product 1 in January 2023, then store 2 and 3 should also have that price for that product for that month.

ChatGPT

**ChatGPT**

To ensure that each store in the country has the same price for each product for a given month, I can help you update your data accordingly. Let's start by loading and inspecting your data to understand its structure and identify where the prices are listed.

I'll load the data from the file you provided, **Complete\_Sales\_Data.csv**, and take a look at the first few rows to see what we're working with.

The data includes columns such as 'Country', 'Store ID', 'Product Category', 'Product ID', 'Date of Sale', and 'Price Sold'. It appears that 'Price Sold' is the column we need to focus on to ensure price consistency across stores for each product per month.

Here’s how we can proceed:

1. **Aggregate Prices**: Calculate the median or mean price for each product per month, across all stores.
2. **Apply Uniform Prices**: Set the price for each store and product to the calculated median or mean price for that month.
3. **Update the DataFrame**: Replace the current 'Price Sold' in your data with these uniform prices.

Would you prefer using the median or mean price for this adjustment? The median is generally more robust to outliers, which might be preferable if the pricing varies significantly among stores. ​​

User

**You**

median

ChatGPT

**ChatGPT**

Great choice! We'll use the median price to ensure consistency. I'll calculate the median price for each product per month across all stores and then update the 'Price Sold' for each entry in your dataset accordingly.

Let's proceed with these steps:

I've updated the 'Price Sold' in your dataset so that each product has the same median price across all stores for each month. This change ensures price consistency as you requested.

Would you like me to save this updated dataset to a new CSV file for you, or is there anything else you need assistance with regarding this data? ​