

|  |
| --- |
| Sales Out Data and Why we collect it. |
|  |
| December 16  Wala Wala Bing Bang  Authored by: Russell Kay |



Contents:

1. Introduction
2. Data source 1
3. Data source 2
4. Data source 3
5. Reading the data into the data warehouse
6. Implications and Value
7. Analysis
8. Viewing the Data
9. Introduction

The data I will be working with is gathered from our customers and shows how much they have sold over the last week. We can also use this data to keep track of stock in their warehouses and the value at which we sold the product to them.

We use the data to see how we can better work with our customers to help them sell the stock we provide. We do this through a range of deals with the customer so that they might be able to sell the product cheaper and hopefully increase sales. We will also look at stock levels to determine how frequent and at what quantity we need to restock the customer.

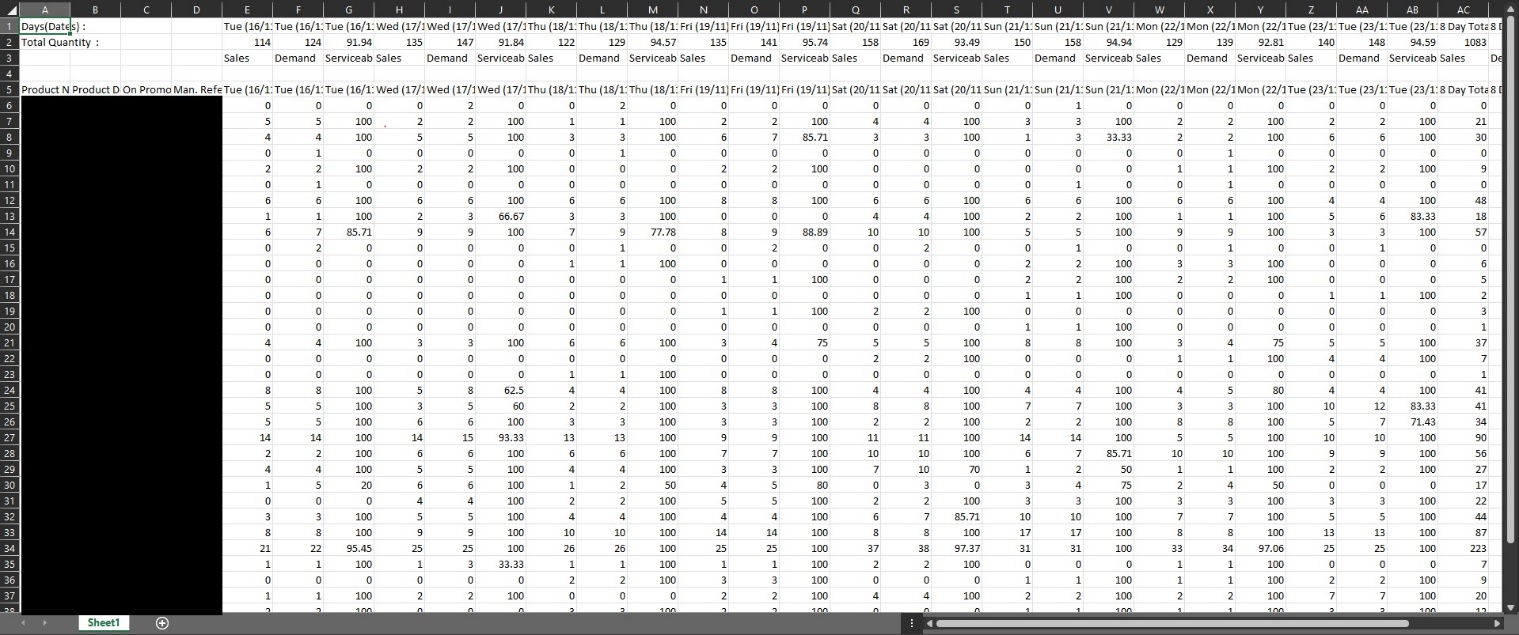
The sales directors also use this information to try and predict next year’s sales and take this into account with their budget calculations.

All in all, this is very valuable data as it offers a lot of insights into what products are selling, where they are selling and what products are not selling. This is so we can decide if it is worth bringing the same product in again whether it be at an increased or decreased amount or sometimes discontinuing the product if sales are poor.

2.0 Data Source 1

Data Source 1 is from Customer 1. To retrieve this data, I need to visit the vendor portal and manually download it and is available every week. This is in a structured .xlsx format and contains quantitative data such as Units sold, and the value of the units sold.

Please see screen shot of data below.



The blacked-out section is the product codes and numbers for the items sold. The data to the right is structured by having the sales date as the column headers and the Qty sold on that date.

At the top you have the sums of the total number of products sold on that date as well as the 8 day running totals and last week’s sales.

This data is retrieved by a script that automatically downloads it from the customers website and renames the file before placing it in the folder ready to read into the Data Warehouse. I manually check the files each week to make sure they are read in correctly.

3.0 Data Source 2

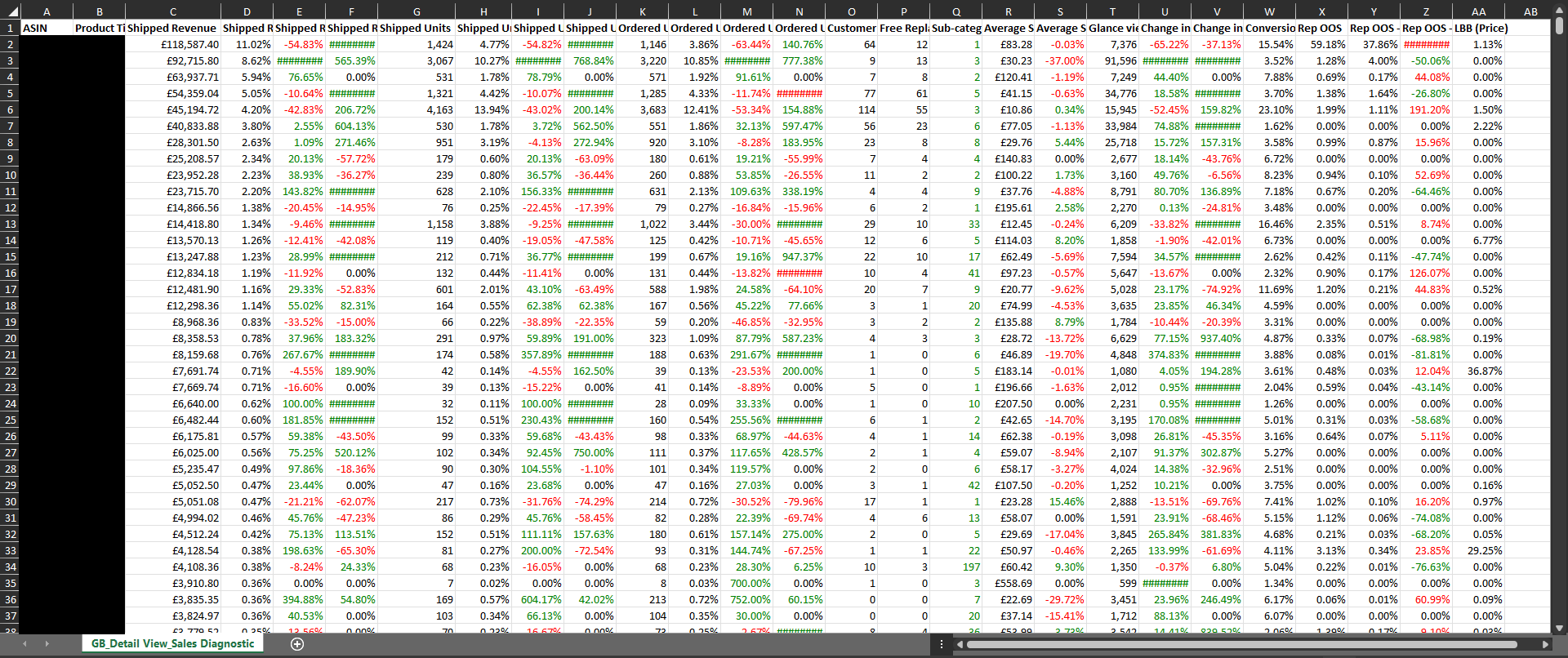
Data Source 2 is from Customer 2. This is also downloaded from the customers vendor portal and is usually available once a week.

This data gives us the total amount shipped over the last week as well as the total revenue.

The data comes in either csv or xlsx however we use the xlsx format. The data is very structured and is simply one table of data. It gives us values we can quantify such as the number of units shipped and the revenue that was made from these transactions.

The data also includes data from the same period in the previous year and the difference between the two.

Below is a screen shot of the data.



Again, the blacked-out section is a list of products with the corresponding values to the right. The two columns we use from this data is the Shipped Revenue and Shipped units.

Shipped Revenue shows us the total amount revenue made by that product and the shipped units shows us how many units were shipped to generate that revenue.

This Data is Quantitative and structured. The data in the workbook is numerical and so easily quantified. Coming in a structured format makes it easy for us to add this data to our data warehouse. It is also external data as we need to manually download it form the customers vendor portal in order to retrieve it.

4.0 Data source 3

Data Source 3 is from Customer 3. This customer provides 2 files. One for sales and one for stock.

The first file is the Sales File. This is a structured .CSV file with quantitative data in a numerical format.

We are emailed this data once a week by the rep who receives it from the customer.

Diagram

Description automatically generated with low confidence

The second file that is received is the Stock File. It is received in the same way and is also quantitative numerical data stored in a structured .csv format.

The Image below shows the structure of the sales file.

Graphical user interface, application, table, Excel

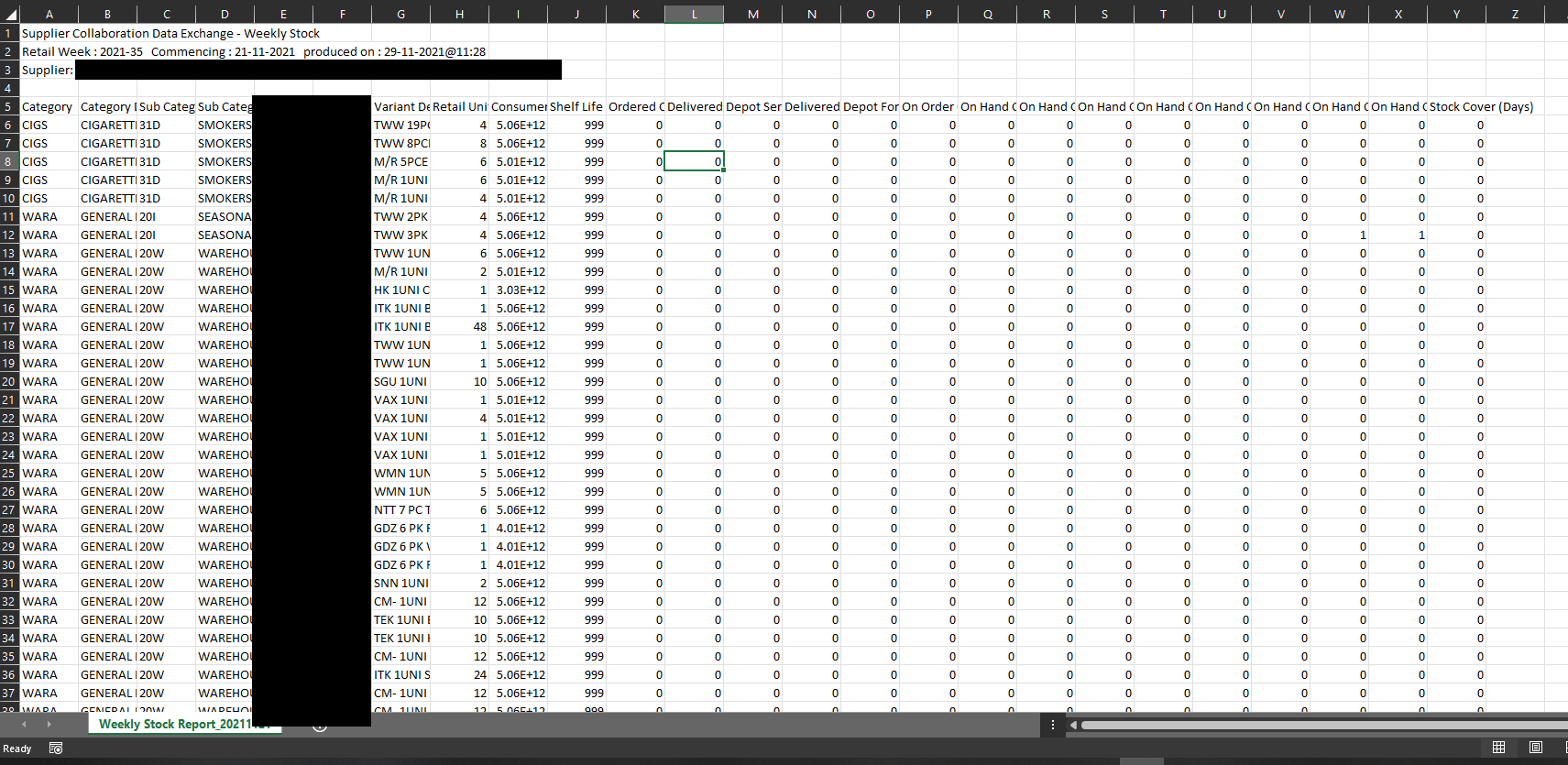
Description automatically generated

The blacked-out column is their product number for the item. We cross reference this with a cross reference table where all the customers Product numbers are stored along with our corresponding Item number.

The two columns we use from this sheet are The Customers Product number and the Sales Units Column.

From the sales unit’s column we obtain the number of units sold in the last week for the specified items on the left.

The Stock sheet is very similar in its structure as shown in the image below.



The two columns we look at in this file are again the customers product number which we reference against our Customer Cross Ref Table. The second is the column Retail Units. This gives us the number of the item the customer has in stock. From this we can look at how fast or slow certain products are moving if we compare it to the Sales File. We can then use this data to try and predict and prescribe what actions should be taken. For example, if an item is selling fast and the customer has a low amount of stock left, we can then predict when we may need to send them more stock and how much stock to send.

All that is needed with these 2 files is to rename them appropriately so that they can be read into by the python Script I Maintain. Once read in they are checked by me manually to make sure the figures read in are correct. (Sometimes the customers have a sneaky habit of changing the layout slightly without informing us).

5.0 Reading the data into the data warehouse

Each of the previous data sources are renamed to a structured format (29-11-2021-CustomerName-WeekNo).

The Date in the file name needs to be the previous Monday’s date. This is when the week begins and because the data is collected weekly it is also used as the date when reading in the data so that we know which week it relates to.

Adding the dash’s makes it easy to extract the text because you can turn the name of the file into an array when the script for that data source runs.

Each of the data sources have a separate python script that reads the filename to get the date and then reads the data itself to populate the table information.

I am currently in the process of updating all the scripts to Python 3 and attempting to make it as standardized as possible in order to make it easier to incorporate any new data sources we may acquire.

The biggest benefit to reading in the data this way is speed. A python script can read the data and then push the data to the data warehouse much faster than a person can and sometimes these files can contain thousands of rows.

Each script also has a built-in alert for if something is wrong with the import. So, the script will run and if it can’t detect the folder or the file it will send an alert to slack which will then send us a notification letting us know it has failed and why it has failed.

Other things are accounted for as well such as Checking the column names to make sure they haven’t changed (Which does happen). And that the data is in the right format.

Each script uses a stored procedure on our SQL server to Push the data into the correct table.

Some values are static such as customer number so this is added by the script. Other data such as Sales units and revenue are taken directly from the file.

The information that is stored in the data warehouse is then pushed to a second data warehouse on Azure. It is transformed and pushed using ETL processes written in SQL. This second data warehouse is used for our reporting in power BI Service. Both on premises and Azure Data warehouses are SQL relational databases.

All the data is loaded into one sale out table that stores all the individual lines from each file that is read in. Blended if you will into one large dataset. Obviously different files offer different data, so Constants are set and used in the python code Where a specific column has the data missing in the file allowing us to create the one table. This is done by using the same stored procedure on the SQL server for each file.

The Below Image Shows the part of the code that validates the file to make sure that the structure has not changed

Text

Description automatically generated

If the file structured hasn’t changed then the below code Reads the data from the file, and populates all the variables for the row data, which is then appended to an array ready to be pushed to the Sales Out Table. See Image Below

Text

Description automatically generated

The next image shows the part of the code that pushes the data to the server.

Text

Description automatically generated

As you can see in the code, we try to capture any error that could happen by using try and excepts. so if it fails at any stage we know where it has failed and can diagnose the problem quicker.

I also use Config Parser module for python. This allows me to store variables and strings in a separate .ini file so the code can be easily modified to accept another data source. File paths and connection strings are 2 of the things that can be stored in the .ini file. I also use logging to store the errors when something breaks as well as pushing a notification to our slack channel which then pops up a notification for us.

Data is restricted in both on premises and Azure tables. For the source system, there are permissions in place so that only those that need to can see the data. Things like ship to addresses are only viewable by sales and customer services in the source system and then only a few people can see list views of the addresses. The Azure and Power BI Desktop can only be viewed by those who have been invited and even then, the row level security means that the reps can only see the customers they are assigned to. The only people in the building who have access to our servers are members of the IT team.

6.0 Implications and Value

The implications of this data are mainly positive. However, this is data we have no control over and is provided by the customer. This means we are reliant on them providing accurate data. There has been at times instances when somebody else’s product has been added to our data, which can cause an issue with reading in the data when it can’t find a reference to the item number.

If this happens it shows unknown in the report. Unfortunately, the python scripts do not detect this when reading in the file as they have no direct reference to check. This data can then skew the rest of the data even if it only be by a small amount.

This has not happened often but when it does, the affected row or rows are removed from the Data warehouses and the source file.

The positive implications (Value), include, providing us with Historical data so that we can perform Descriptive Analytics by looking at the data and seeing where products sell well and where they don’t. We can then take this data and diagnose what went wrong. Was the item sold at the wrong time of the year? Is the pricing off? We can then take that information and try and predict what are sales will be like in the future by looking at various periods in time. We will also use prescriptive analysis with strategies to try and avoid what may have not worked so well in previous periods or even a way to bring down costs. One example of this is predicting how much stock a customer will need in their warehouse, so that we do not provide them with too much or too little and in doing so can save on storage and transporting costs.

We can also use the data to compare our customers to see which one is moving more stock and those that we could potentially try and sell more to. We can also see which product categories are being sold amongst each customer. Sometimes we will do deals with the customer on some under performing ranges by reducing the cost at which they pay for the item. Sometimes small rebates are given if they sell the stock.

From this data we can also see the trends of product sales through out the seasons and try and predict when we will bring more stock back in for that product in an order for us to keep our costs down by not storing the items longer than we need to. Or even if its worth us bringing in any more stock. If a product isn’t selling, we are not going to bring anymore back in.

7.0 Analysis

In this report so far, we have looked at the data sources, how they are read into the system and the implications and value to the company.

As described all the data sources are structured data that can be easily quantified. This means we can sum and group by customer, item, IPG, Category, Brand, Sales Rep, and company. This is because of relationships already set up in the data warehouse between the relevant tables. The data is usually in .xlsx or .csv. Both are easy to read straight into a database.

The data offers us a way of describing what has happened over a certain period. The longer the period the bigger the picture. This then allows us to diagnose what factors may have affected sales whether they be good or bad. It also gives us a basis when working out the budget for the following year and where we need to focus more effort in order to grow.

We have also discussed how the data goes through stages of being read in and then transferred to our cloud Data Warehouse on Azure. This is where the data comes to life.

After the files have been read in the python scripts move them to an archive folder so that we have a record of the original data. The script will move the file to the error folder if something goes wrong as well as a notification to slack of what went wrong with the import.

In Values and Implications, we discussed the benefits and drawbacks of using external data.

The biggest drawback is not being in control of the data that is produced. However, we do not have a way of producing that data ourselves.

The data can also offer a lot of value to both us and the customer by allowing us to supply the right product to the right customer at the right time so that costs can be reduced by both us and the customer. To do this the sales directors will use prescriptive and predictive analysis and factor these figures in when working on next year’s budget. External Trusted open Data Sets are of great value to the customer and us. It also connects us to the digital world allowing us to directly access this information from the customers portal.

All of these portals are accessible through Portable devices such as phones, tablets laptops. All also have the capacity to forward that information on.

It also allows the company to set targets for the reps to meet based on there previous years data. Usually, bonuses are calculated from this.

In its current format the data isn’t very readable by a person even though it is structured. There are a lot of rows and a lot of data to take in. Making the data more readable to people and more easily absorbed is through visualizations and reports. We achieve this by using a service call Power BI. Which leads us into our next Section.

8.0 Viewing the Data

To view the data, we use the Power BI service and an Azure Data Warehouse to store the data. The data warehouse is not just for sales out data and includes information on Sales, invoices, and forecasts as well as others.

Once the sales out data has been read in and the report refreshed. It can be viewed by those with permission to do so. Reps can only see the customers that relate to them.

The sales out data report has 2 views. One for value and one for units out. These are seen in the below screen shots.

The first screenshot is of the sales unit’s view. The blanked-out sections are the filters for each customer. In the top right-hand corner of the report is a switch that when pressed shows the values.

A picture containing table

Description automatically generated

The second screenshot is of the Sales Values View.

Table

Description automatically generated

We have a timeline filter so that quantities and values can be checked over a specific period by simply dragging the timeline to where you want it. You can also change the timeline from years to months to weeks and to days.

The report Use’s both a matrix and visuals to present the data. The visual is a simple Bar chart showing the amount out last year (Value or Units), compared to the amount this year. There is also a line in the chart showing the level of stock with customer during the selected period.

The Matrix shows the actual figures for value and units out In the Amount Out column there is a small visual indicator showing if sales were up down or remained the same for a quick reference. This is displayed as an arrow as seen above.

The matrix also has multiple levels that you can drill into. The below screenshots show examples of the different levels in the report.

Week Number.

Graphical user interface, text, application, chat or text message

Description automatically generated

Brand.

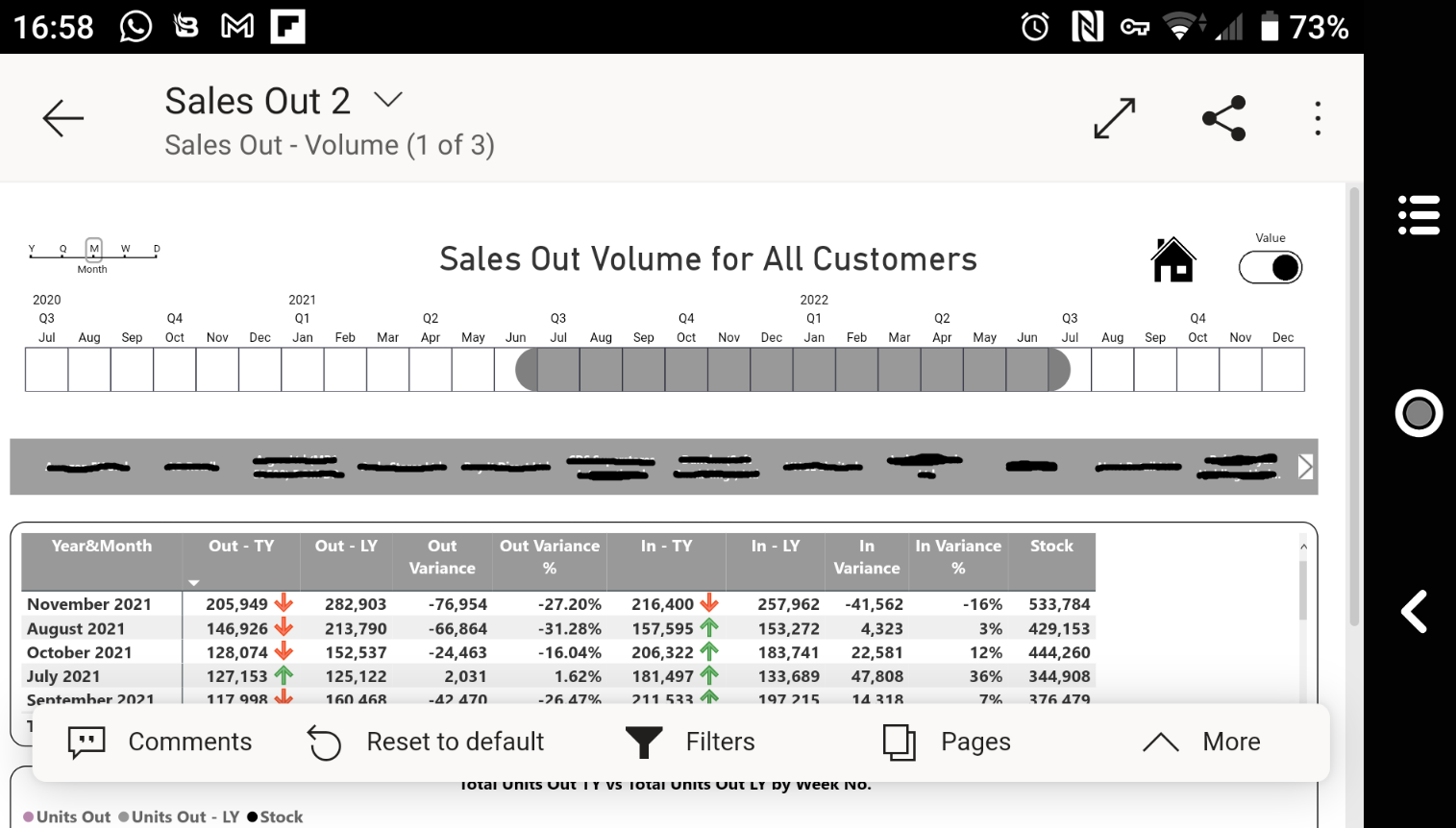
A picture containing text

Description automatically generated

Other levels include, IPG, Range, Item, and customer. This allows us to analyze the data from different directions.

We also make use of the Internet of things by making the reports available on mobile devices.

Below is a screenshot of the report on a mobile device.



This offers all the functionality of using it on a computer with the added benefit you can take it with you.

Using the Internet of Things, we have made our data available whilst the reps are out on the road. Power BI Service can be accessed through Both the google play store and the Apple App Store.

Visuals are an effective way of showing whether sales have increased or decreased and in the above reports you can see that there are visual indicators in the TY (This year) columns that clearly show with the use of arrows either up or down. As well as the graph below the matrix.

As you can see from the visuals, we collect data from a lot of different customers. The 3 selected in this project are good examples of how all the data we collect is structured. They are always either structured .CSV or .xlsx (.xls sometimes). At one point one of our customers provided info on comments about our products which would have made a good example of qualitative data, however they dropped this service earlier in the year.

Even though our data is collected in a structured manner we do also have the means to make use of semi structured data such as JSON’s, .xml etc.