Market Basket Analysis

Kostas Voulgaropoulos

3/5/2021

Overview

The following report showcases a market basket analysis and its use to a retail store. Market basket analysis helps identify subtle connections among products and can be found extremely useful in product placement and marketing discount offerings.

The data used for this certain example is public, donated in 2015 to the UC Irvine Machine Learning Repository and can be found here.

From the repository's description: *This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.*

After a basic preprocessing our dataset looks like this:

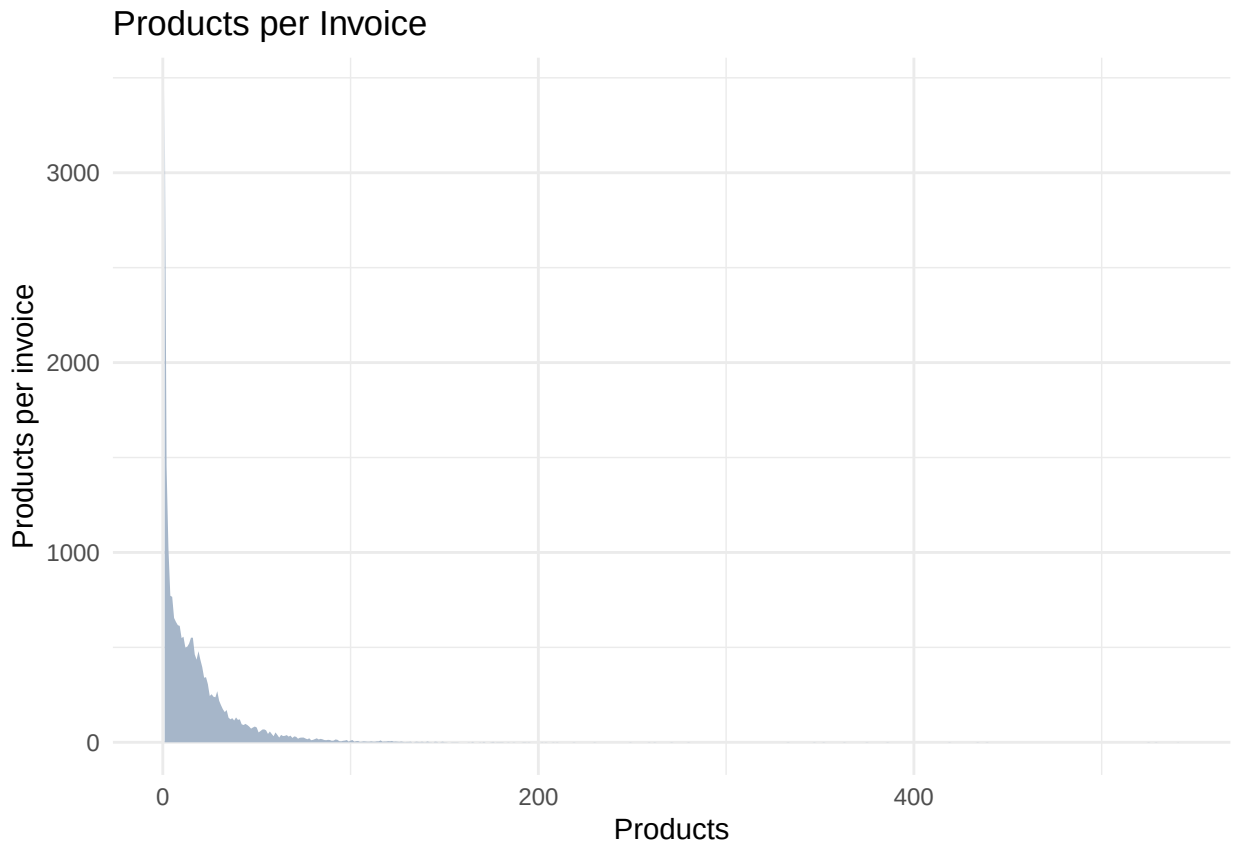
```
## # A tibble: 6 x 10
##   InvoiceNo StockCode Description         Quantity InvoiceDate         UnitPrice
##   <chr>     <chr>     <chr>                  <dbl> <dttm>                  <dbl>
## 1 536365    85123A    WHITE HANGING HEAR~        6 2010-12-01 08:26:00      2.55
## 2 536365    71053     WHITE METAL LANTERN        6 2010-12-01 08:26:00      3.39
## 3 536365    84406B    CREAM CUPID HEARTS~        8 2010-12-01 08:26:00      2.75
## 4 536365    84029G    KNITTED UNION FLAG~        6 2010-12-01 08:26:00      3.39
## 5 536365    84029E    RED WOOLLY HOTTIE ~        6 2010-12-01 08:26:00      3.39
## 6 536365    22752     SET 7 BABUSHKA NES~        2 2010-12-01 08:26:00      7.65
## # ... with 4 more variables: CustomerID <dbl>, Country <fct>, Date <date>,
## #   Time <chr>
```

Exploratory Data Analysis

First, we need to explore a little bit our data so that we get a feeling of what to expect. In the EDA section, we seek to answer generic questions about our dataset.

How many different items do customers buy during each visit?

In case most of the times our customers buy only one thing during their visit, a market basket analysis is not only difficult to perform (since associations exist in **groups** of products) but also **without much value**.



What we see from this graph is that:

-The **most frequent** number of items per invoice is **1** (3434 Invoices) or at least close to 1

```
## .
##    1    2    3    4    5    6    7    8    9   10
## 3434 1453 1020  772  766  655  634  617  613  549
```

-There are **extreme orders**, even an invoice with 541 (!) items

```
## .
## 352 363 375 386 419 434 439 525 529 541
##   1   1   1   1   1   1   1   1   1   1
```

Which makes sense, since the shop also **deals with wholesale**. Wholesale buyers, especially those buying such a variety of products, may not be so representative of the retail scheme. They seek to **maintain a stock**, thus their invoices **do not reflect their own preferences**. But on the other hand, their purchases **reflect** their **customers' desires**, especially if we focus on the volume of the products they buy. They must be conservative on the lower-selling products and bold on the higher-selling.

Overall, there are plenty of invoices that contain more than one item, thus it is **possible** to conduct a **market basket analysis**.

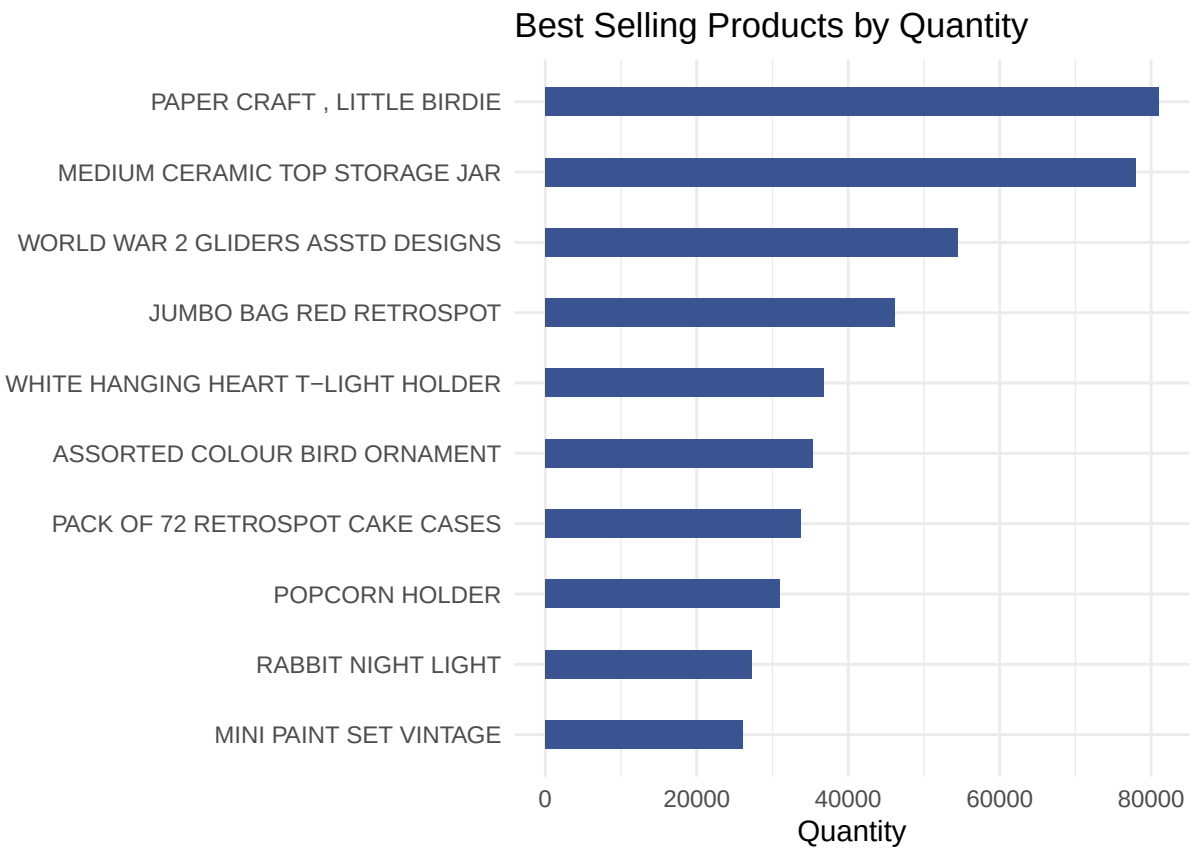
But the fact that 1 is the most frequent number, highlights the need for some extra analysis on those orders.

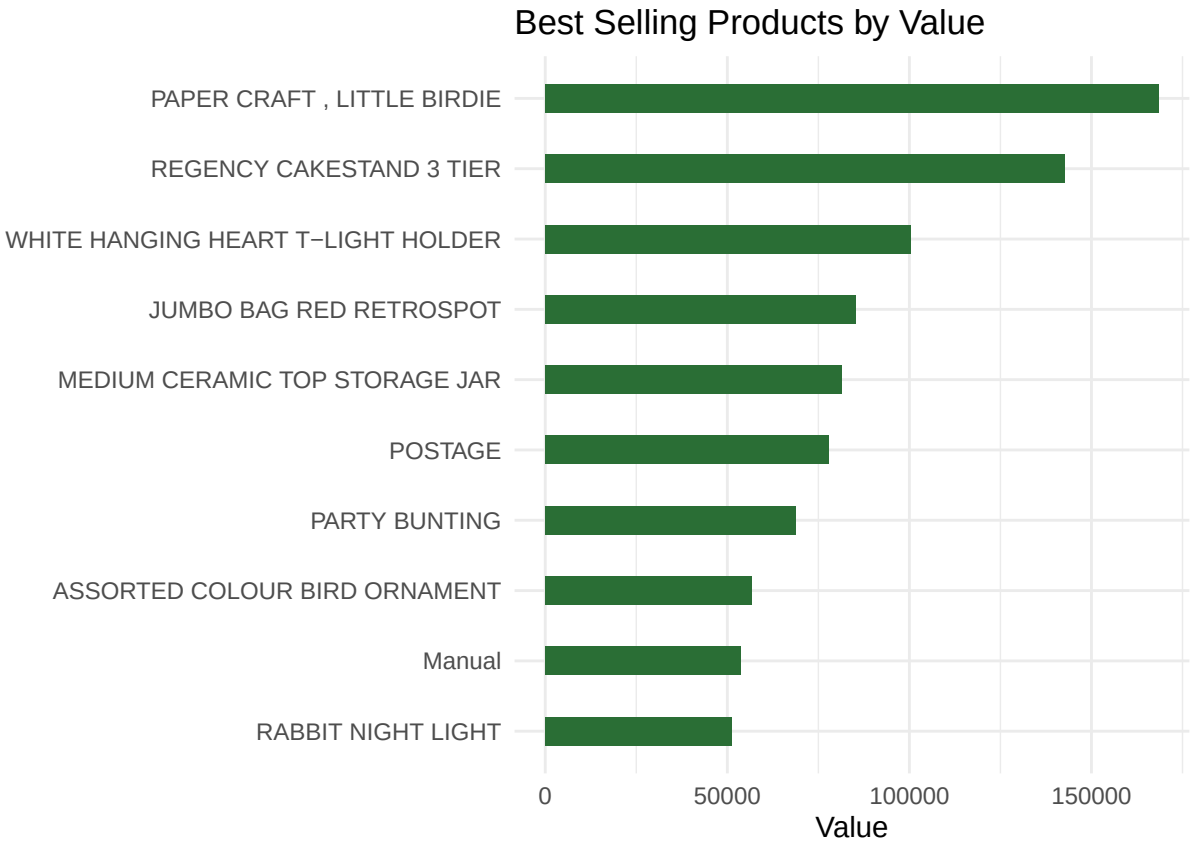
Best and Worst selling products (by both quantity and value) and how often are they purchased?

Before we seek to answer these questions, we need to **exclude invoices that represent returns** of products. If we omit this step, the worst selling products might just become the most frequently returned.

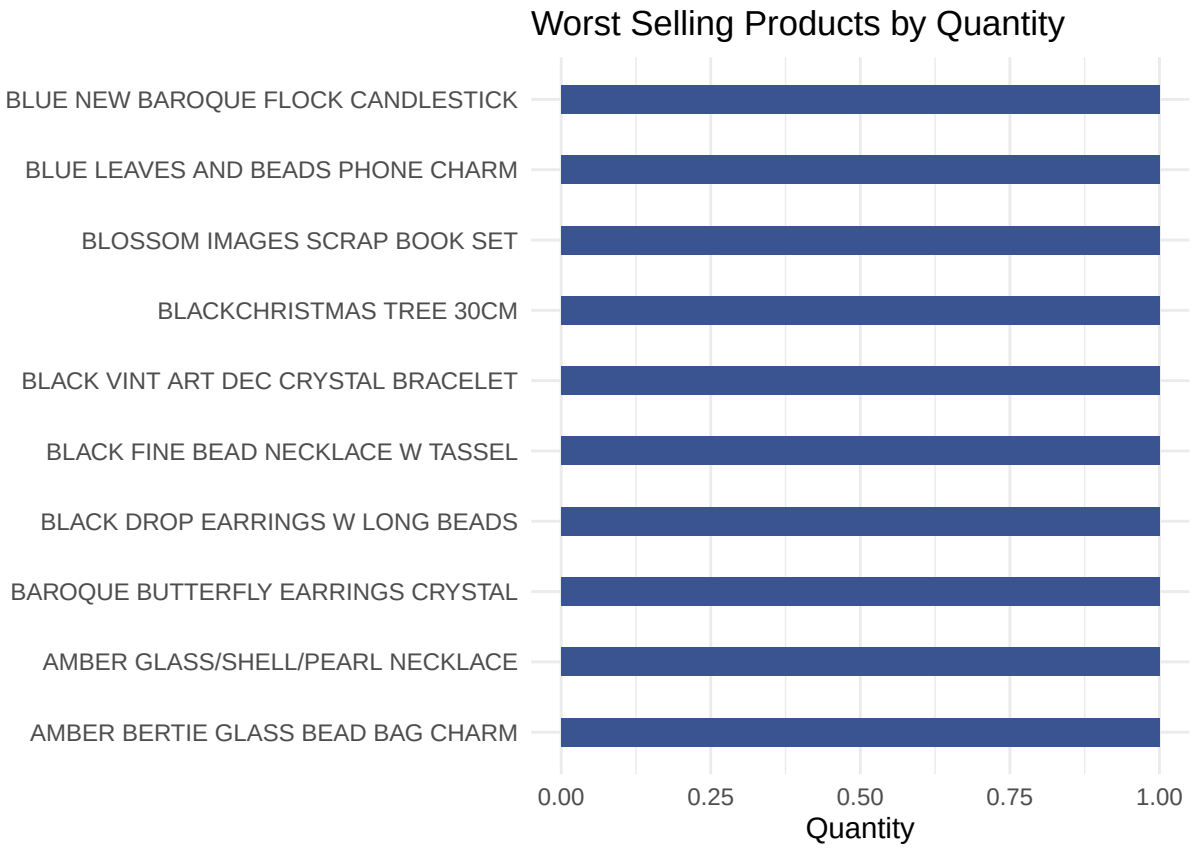
Such data would be valuable for analyzing returns (e.g. detecting defective products), but the scope of this current analysis is limited to sales.

Best Selling





Worst Selling



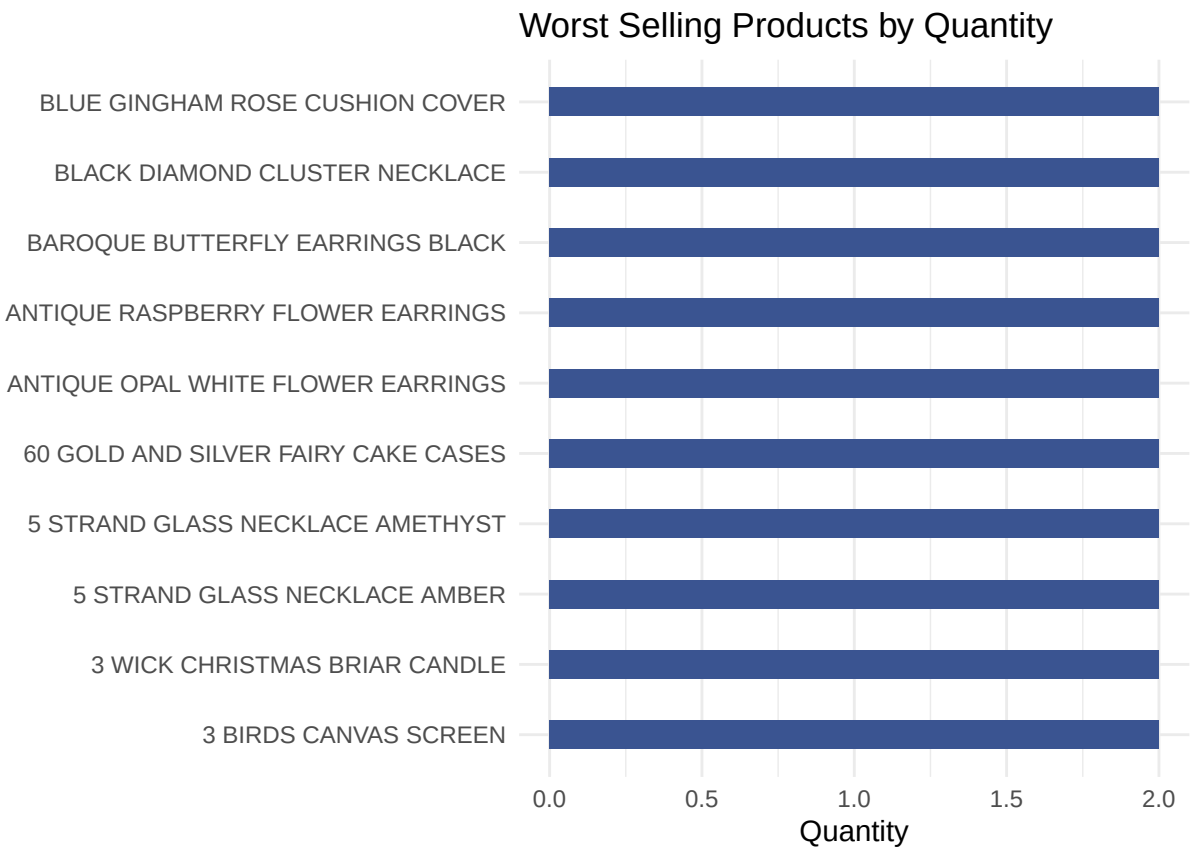
All Bottom 10 Products have only **sold one piece** over a period of years .

How many products have this fate?

```
## [1] 59
```

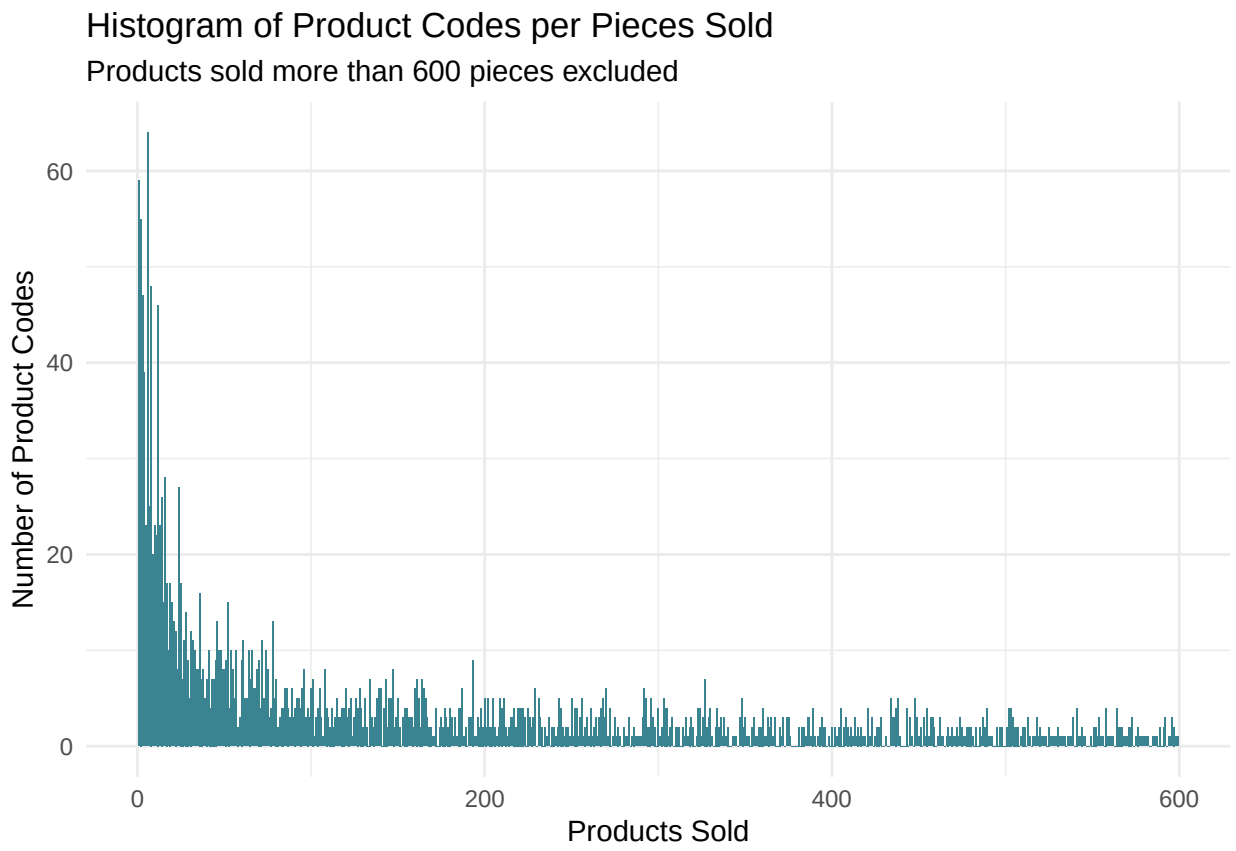
59 distinct products have moved only once! Pretty useful information, especially if we seek to optimize workload exercise.

So, excluding all those laggers, the above graph might look like:



Now we're stuck in the neighborhood of **2 moves** over the course of those years.

Perhaps a density plot could show us what to expect in the aforementioned neighborhood of low selling products:

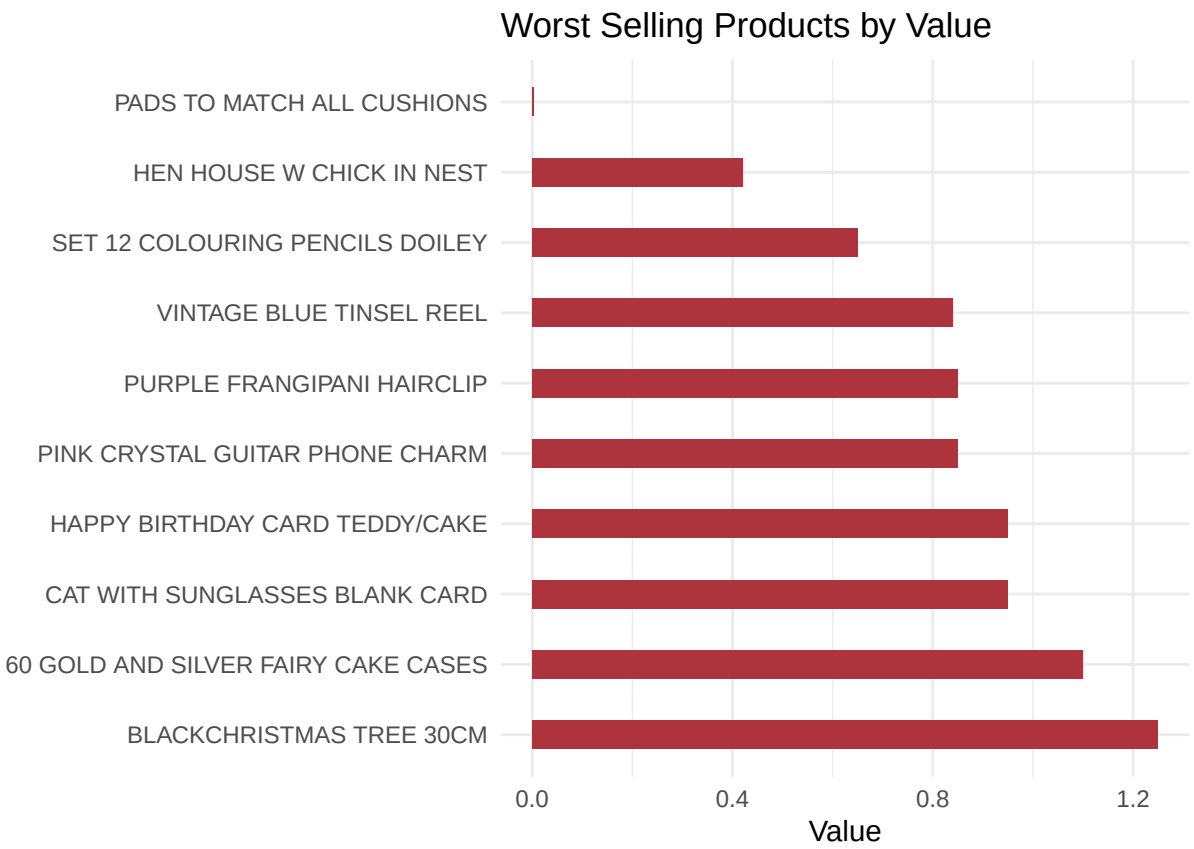


And bingo! There are way too **many slow moving** products.

59 distinct codes have sold only 1 piece, 55 codes sold 2 pieces, and the list goes on and on.

Unfortunately, with such measures, looking at the **10 lowest selling products** is of **no value**, since the low selling are way more than 10.

Let's now look at the worst value bringers:



There are products that have generated **less than 1 pound of revenue over years**. Perhaps a retailer might gain insights from that.

Though it might not be of obvious use, knowing the best and worst sellers can be effectively **combined** with a market basket analysis and **affect the marketing strategy** of each retailer.

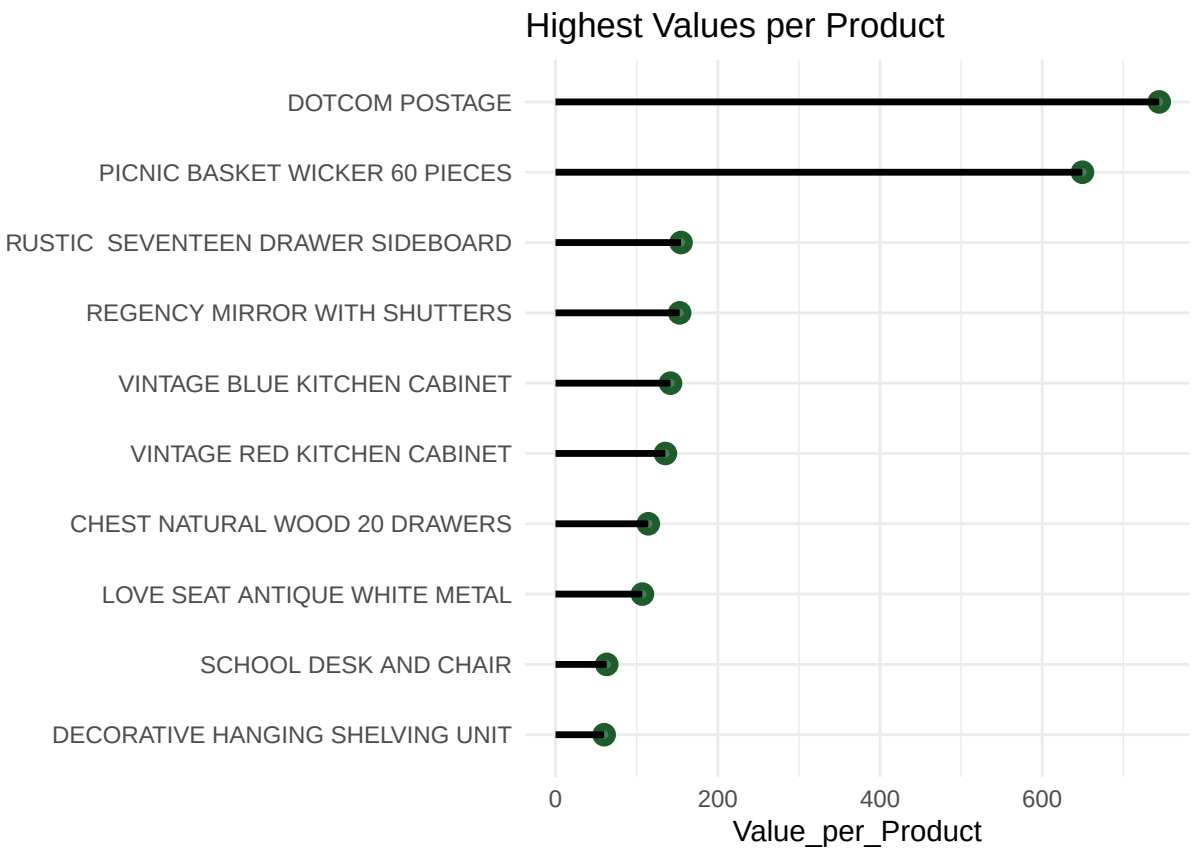
For example, a well performing product might prove associated with both one well and one poorly performing one. The retailer might decide to connect it with the poorly performing in order to **boost its sales** or totally **abandon** the hopeless poor performer and target the **boosting of the two well performing** products by combining them.

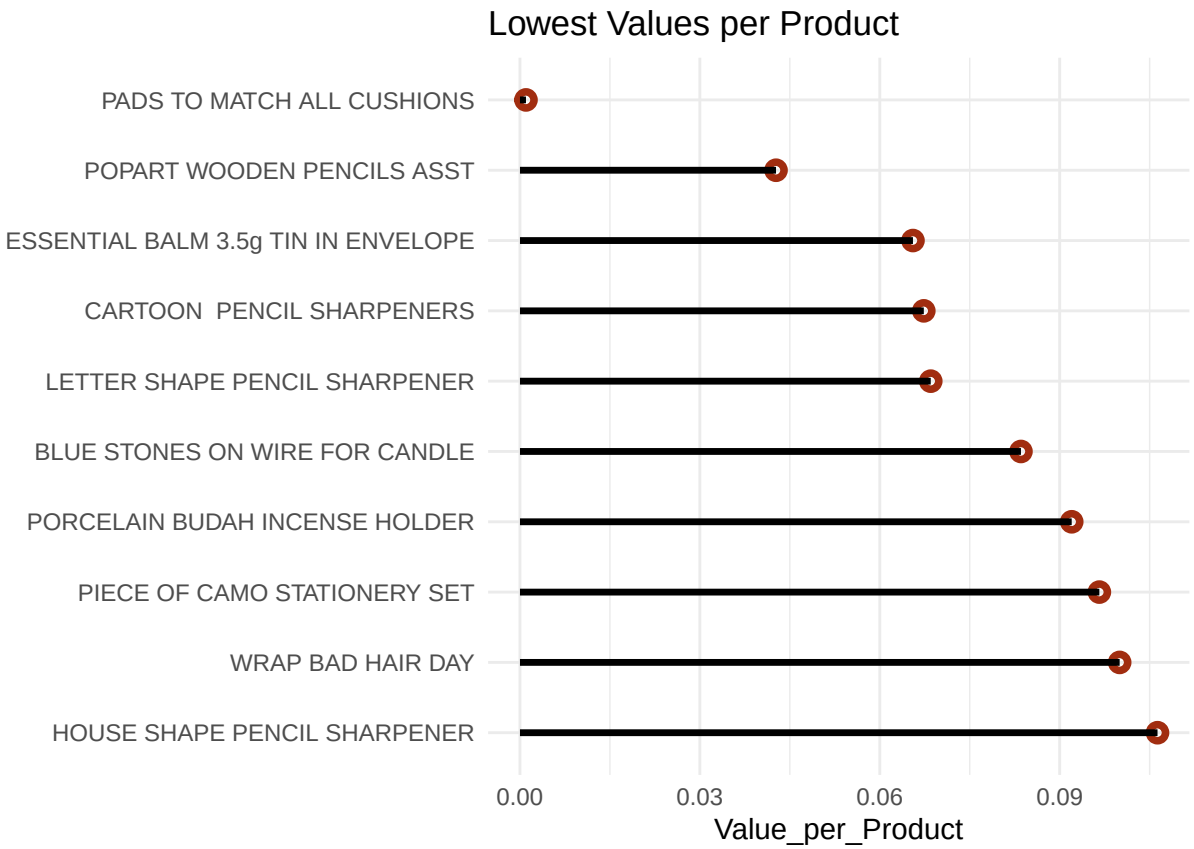
One way or another, the overall performance of a product is useful as a context in making decisions from a market basket analysis.

Products of Highest and Lowest Effort

Expanding the value of context, a retailer will surely find useful knowing which products generate revenue without needing much warehouse circulation (and thus effort) and which don't.

That's why a look at the average revenue per product would be insightful.


Highest Values per Product
DOTCOM POSTAGE
PICNIC BASKET WICKER 60 PIECES
RUSTIC SEVENTEEN DRAWER SIDEBOARD
REGENCY MIRROR WITH SHUTTERS
VINTAGE BLUE KITCHEN CABINET
VINTAGE RED KITCHEN CABINET
CHEST NATURAL WOOD 20 DRAWERS
LOVE SEAT ANTIQUE WHITE METAL
SCHOOL DESK AND CHAIR
DECORATIVE HANGING SHELVING UNIT
0
200
400
600
Value_per_Product



The above insight can be evaluated in context with the results of the market basket analysis.

A store owner could for example choose to prioritize rules that include low effort products over others with higher effort products.

But enough with the Exploratory Analysis and the Graphs, let's go on with mining association rules.

Market Basket Analysis

Market Basket Analysis is based on **Association Rule Learning**, a machine learning technique that helps us identify rules in large datasets.

An example of an interesting rule (expressed in the form of if product X is then Y is also bought) is the "Beer and Diapers Correlation" identified in the early 90's.

Even though it hasn't reached consensus, the most prevalent explanation of this rule is that fathers that to go buy diapers for their children in the evening after work, reward themselves with a six-pack of beers.

In the present analysis, we will use the **Apriori Algorithm** to mine associations hidden in our dataset.

Apriori calculates **certain parameters**, which are better understood with the following example:

Let's say a sports store has: 100 transactions (or invoices) in total, 13 of which contain a tennis **racket**, 12 contain tennis **balls** and 10 contain **both** a racket and balls

Between the tennis racket (product A) and the balls (product B), there is:

1. **Support** of 0.1 or 10% (= 10 / 100), which represents the **Frequency of the combination** of interest (10) divided by the **total transactions** (100).

2. **Confidence** of 0.77 or 77% (= 10 / 13), which represents the **Frequency of the combination** of interest (10) divided by the **Frequency of Product A** (13)

3. **Lift** of 6.41 (= 0.77 / (12/100)), which represents the increase in the ratio of sale of B (12/100 in our example) when A is sold. In other words, lift shows that the **likelihood of buying** a racket **and** balls **together** is 6.41 times more than buying the **balls alone**. Lifts greater than 0 signify that the two products are **more likely to be bought together** than separately, and vice versa (Lift < 0 shows that separate sales are more likely than those of the group).

4. **Coverage** of 0.13 (= 0.1 / 0.77 or 13/100). Coverage is also called left-hand-side support and represents a measure of **how often** the rule can be applied. In practice, it shows us **how often product B appears**.

So, before we train our APRIORI algorithm, we need to specify which kind of rules we would like to see.

In this certain analysis we will be strict with our rules and choose to only see rules with:

1. **Support** greater than 0.0095 or 0.95

2. **Confidence** greater than 0.9 or 90

3. **Maximum Length** less or equal to 8. That means we don't care for combinations that contain more than 8 products.

Now that we have tuned our algorithm's parameters, let's go on and observe results:

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5  0.0095      1
##  maxlen target  ext
##       8  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 176
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[3866 item(s), 18566 transaction(s)] done [0.15s].
## sorting and recoding items ... [656 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 done [0.02s].
## writing ... [7 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].

##     lhs                                    rhs                                    support confid
## [1] {REGENCY TEA PLATE PINK}            => {REGENCY TEA PLATE GREEN}           0.010880103  0.901
## [2] {REGENCY TEA PLATE PINK,
##      REGENCY TEA PLATE GREEN}           => {REGENCY TEA PLATE ROSES}           0.009964451  0.915
## [3] {REGENCY TEA PLATE PINK,
##      REGENCY TEA PLATE ROSES}           => {REGENCY TEA PLATE GREEN}           0.009964451  0.939
## [4] {SET/20 RED RETROSPOT PAPER NAPKINS,
```

```
##      SET/6 RED SPOTTY PAPER CUPS}         => {SET/6 RED SPOTTY PAPER PLATES}      0.009533556  0.921
## [5] {POPPY'S PLAYHOUSE BEDROOM,
##      POPPY'S PLAYHOUSE LIVINGROOM}        => {POPPY'S PLAYHOUSE KITCHEN}          0.010018313  0.907
## [6] {WOODEN HEART CHRISTMAS SCANDINAVIAN,
##      WOODEN TREE CHRISTMAS SCANDINAVIAN}  => {WOODEN STAR CHRISTMAS SCANDINAVIAN} 0.009802865  0.910
## [7] {REGENCY CAKESTAND 3 TIER,
##      ROSES REGENCY TEACUP AND SAUCER,
##      PINK REGENCY TEACUP AND SAUCER}      => {GREEN REGENCY TEACUP AND SAUCER}    0.012872994  0.901
```

And Voila!

These **7 combinations** satisfy the criteria set earlier.

We can see for example that:

1. REGENCY TEA PLATE PINK is combined with REGENCY TEA PLATE GREEN in 1.09% of the transactions.
2. 90.18% of the times REGENCY TEA PLATE GREEN was bought, REGENCY TEA PLATE PINK was also bought
3. REGENCY TEA PLATE GREEN is present in 1.21% of total invoices
4. it is 62.01 times more probable for REGENCY TEA PLATE GREEN to be **bought along with** REGENCY TEA PLATE PINK, than without it.
5. There have been 202 invoices that contain both REGENCY TEA PLATE PINK and REGENCY TEA PLATE GREEN

Time Series Decomposition

Before we sum up, let's take a look at the value of **sales as a time series.**

That means to disregard all features of our dataset and concentrate on one numeric of our choice (Value of Sales here) and analyze it through time.

Time series decomposition is a technique that **decomposes a time series of values** (e.g. Sales of Air Tickets, Weather Data, etc.) into:

1. Seasonal Fluctuation
2. Trend
3. Random Noise

That decomposing allows us to **spot the trend** (that's the main point of interest) and distinguish whether it is ascending or descending. By figuring out the trend's momentum, we can project that to the future and gain an overall view of the future (that is solely based on the past, no external factor is taken into account).

Unfortunately, for this technique to work properly, we need: 1. our data to cover a large timespan 2. a seasonality to be present

A nice example of a working case is the Airline Passengers.

This certain dataset contains more than **10 years** of observations and **trend and seasonality** are so prevalent that they are obvious even to the naked eye. At first glance, it seems that there is a **yearly pattern** with high summer and low winter numbers that is **repeated in higher values** every year.

Our data contain a little more than a year of observation, thus it would be nearly **impossible to capture yearly fluctuations** (for example high Christmas sales, low sales while people are on vacation, Black Friday frenzy, etc.).

Missing the yearly seasonality will lead to a **wrong detection of trend** and a wrong prediction as a result.

But we can explore saled thoughout time and look for insights.

First, for each day separately:

We can spot a mildly **ascending trend**, but the daily values are way too many and the line becomes too **sharp for the human eye**.

Perhaps a weekly average would work better:

And now the trend becomes more obvious. Higher values happen in the upper right quartile of the graph. One could consider it proof of an ascending trend. But that could also be misleading.

If we combine the high values of late 2011 with the highs of late 2010, we could also attribute the rise in the christmas holidays. Had we had more data, we would be able to distinguish whether this raise happens due to Christmas or due to a rising trend (because we would compare same days among years).

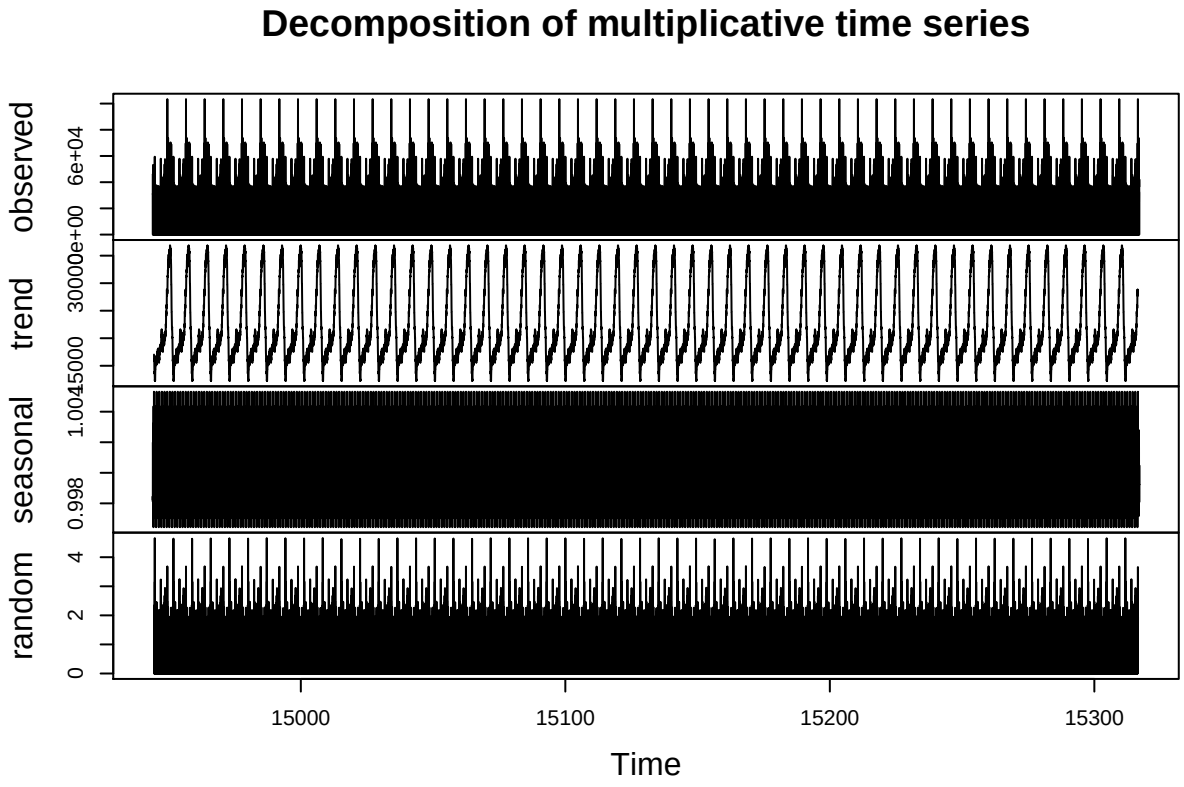
But apart from the yearly seasonal fluctuations, some weekly fluctuation might also be present in many datasets. A peek into the sales of each weekday would enlighten us:

Apart from the fact that there is **no Saturday sale** in the whole dataset, there seems to be **a pick in Thursdays** and an obvious **fall in Sundays**.

The fluctuation shown in the above diagram is a sign that our dataset's sales probably contain weekly seasonality.

We could attempt a time series decomposition with a weekly seasonality, but still the results would not be that much reliable. Plus, values in Saturday will be imputed with 0.

Such a decomposition would look like:



Which appears **very bad** since **we only have one year of data**. A decomposition is obviously of no use here.

Conclusion

We've come to analyze our dataset and, apart from the insights gained from the Exploratory Data Analysis with the graphs, we managed to extract rules useful for **marketing strategy** and **product placement**.

Such analyses are **sensitive** to each user's **preferences** and **goals**. Our tuning parameters (minimum support and confidence) were more or less picked by chance.

On top of it, the few (with such strict criteria) rules that we unearthed, we did not convert them to actionable advice. This is due to the fact that such information needs to be filtered under certain **business criteria** and **priorities** that vary from retailer to retailer.

For example, one retailer may seek to **minimize their warehouse cost**, another one to **maximize revenue**, another to **optimize orders' volume and frequency**, another to **attract new customers**, etc. What is useful to the one, might be meaningless for the other.

Rules exist and each recipient **utilizes** them in a **manner** that **suits them best**.

As for the time series decomposition, **observations do not cover a large timespan** and thus a decomposition would not be so valuable.