Data Analysis Report

Authors: Mukesh Manral, Rafal Pikula, Butan Mangu

|  |
| --- |
| **Purpose of this Data Analysis:** To report on the findings of analysis of spending opportunities that has been run for the procurement function of a company which is a Poultry Farm |

Table of contents

[1. Introduction 3](#_Toc84069745)

[1.1 Purpose of the Analysis 3](#_Toc84069746)

[1.2 Main Assumption and Goal 3](#_Toc84069747)

[1.3 Main Findings 3](#_Toc84069748)

[1.4 Key Documents 3](#_Toc84069749)

[2. Data Selection and Approach 4](#_Toc84069750)

[2.1 Recommendation 10](#_Toc84069751)

[3. Discussion 11](#_Toc84069752)

# Introduction

This report outlines the findings of Spent data analysis that has been performed as part of the AV capstone project.

## Purpose of the Analysis

To identify some scope for cost efficiency and better strategic planning in order to generate savings.

## Main Assumption and Goal

We concentrate our efforts on the data from the most recent year provided, i.e., the year 2019, and we try to identify the high-value and high-moving products to look for saving opportunities among them.

## Main Findings

We isolate a group of 11 high-value high-moving products and observe that their prices are the lowest just at the beginning of the year and then they increase, sometimes substantially, and do not come back to the former level. We can take advantage of this generally upward price trend and under some extra assumptions we are able to propose savings in the ballpark of **3.25%** of total spending for 2019.

## Key Documents

We are using the KPMG\_Data\_Spend Analytics 2.xlsx file provided for this project.

# Data Selection and Approach

The data originally contains 75349 rows and 65 columns and has no duplicates. There are 18 columns containing missing values and in several cases the missing value count exceeds 90%. Missing value findings are summarized below:

NaN percentage threshold: 0%

Number of columns with more than 0% of missing values: 18

Missing Data %

Columns with missing values

S.1 99.19

Spec. Stk Valuation 97.27

S 97.27

Cns 95.14

A 95.14

TrackingNo 93.46

Requested By 91.84

Priority 80.56

Input Tax Credit 79.29

Un 59.96

SLoc 17.03

DCI 10.29

NCM Code 2.56

Profit Ctr 2.12

MTyp 2.12

BUn 2.12

Material 2.12

Price Date 0.37

Moreover, the observation that 20 columns contains just 1 value (not including the missing values) has prompted us to have a closer look at the column values for each column available. It turns out that many of the columns are constant or almost constant and hence have no value to us. In addition, many other columns contain information not needed for the purpose of our analysis, e.g., units of measurement or names of people involved in the ordering process. We drop all such columns and we keep the following columns:

'CoCd', 'Material', 'Short Text', 'Plnt', 'SLoc', 'Matl Group',

'PO Quantity', 'Net Price', 'Net Value', 'Gross value', 'Price Date'

We further restrict our attention to the data from 2019 only as this is the most recent year available and we retain almost 90% of all the records and over 93% of total spending:

Percentage of retained records for 2019: 89.573%

Percentage of total spending generated in 2019 only: 93.416%

In this smaller dataset we still have some missing values but to the lesser extent:

NaN percentage threshold: 0%

Number of columns with more than 0% of missing values: 2

Missing Data %

Columns with missing values

SLoc 15.82

Material 1.85

Now, we are ready to look for the savings opportunities. First, we are going to consider cumulative spending (gross value) in order to find products that generate a lot of spending (high-value products). We are going to group by the data with respect to the company code, material and short text columns, where the first two are mostly for the referential purposes.

At first glance we detect a potential issue pertaining to missing values in the SLoc column (storage location) – after computing the cumulative spending with respect to the storage locations only about half of the spending is accounted for:

SLoc CoCd Material Short Text

9101 9000 910970.0 Soya Bean - (MP) 0.042242

910070.0 Soya Bean - (A) 0.079732

910000.0 Soya Bean 0.097162

F001 7860 7101320.0 IB Ross Broiler Finisher Feed 0.113856

F003 7860 7101320.0 IB Ross Broiler Finisher Feed 0.125731

...

9409 9000 950020.0 Gunny-Bags Damage-2 Katta-0.7Kg 0.481437

Gunny Bags-Pp-0.3Kg 0.481437

9403 9000 950020.0 Gunny-Bags Damage-Pp-0.3Kg 0.481437

CGEN 9000 7400000.0 LPG Cylinder 35 Kg 0.481437

9403 9000 950020.0 Gunny-Bags Damage-2 Katta-0.7Kg 0.481437

However, we will see in the Discussion section that this issue is not significant enough to impact our analysis.

Moreover, apart from a handful of cases the description in the Short Text column can be used as a subdivision of the material designation and so after the aforementioned aggregation restricted in such a way so as to retain exactly 93% of total spending (and ignore low-value items) we get 1126 products that we will use as a basis for further explorations:

CoCd Material Short Text

7860 7101320.0 IB Ross Broiler Finisher Feed 0.099324

IB Ross Broiler Starter Feed 0.172742

9000 910970.0 Soya Bean - (MP) 0.221110

7860 7101330.0 B4 IB Ross Feed 0.265036

9000 910070.0 Soya Bean - (A) 0.304566

...

4500 819240.0 Rotary Hammer 0.929993

9000 980250.0 Tesioning Element 0.929995

964930.0 Numex Batch Coder 0.929996

7860 807330.0 Cement Sheet 2.5 Mtr 0.929998

9000 802310.0 MS Angle 65X65X6, 5.80 kg / mtr 0.930000

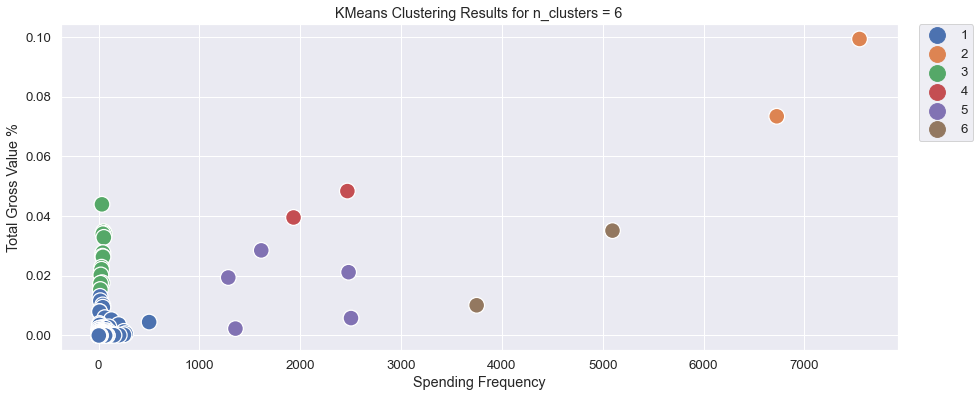
Name: Gross value, Length: 1126

Now, we are going to use clustering to help us select the high-value high-moving products. To this end we prepare a new data set. This new data set contains the Total Gross Value % column taken from the aggregated data as well as two newly computed columns: Spending Frequency (indicating the number of purchases in the given period) and Spending Days (indicating the number of days on which purchases occurred in the given period).

Despite the elevated correlation between the Total Gross Value % and Spending Frequency columns (at the level of 0.75) we select these two columns as the only columns to be fed into the clustering algorithms. We consider the following algorithms: KMeans, Affinity Propagation, Mean Shift, Spectral Clustering, Agglomerative Clustering, and DBSCAN. Moreover, in each case we always preprocess the data using standard scaler with the default settings. The following is the summary of how well these algorithms corroborate our business intuition.

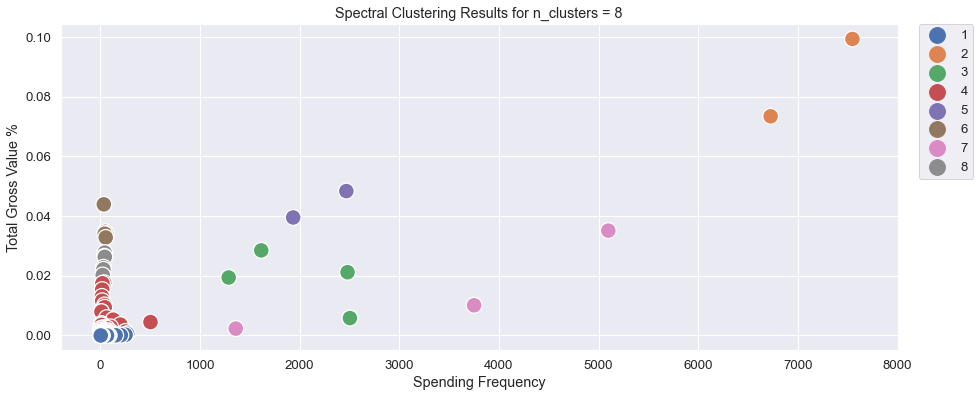
**KMeans**

This algorithm (run mostly on default settings with n\_clusters ranging from 1 to 9 and random\_state set to 7532) is not ideal (requires manual cluster merging), as reflected by the clustering results, since in our case there is a huge concentration of points with relatively low value and low frequency and only a handful of points with relatively high value and high frequency. The number of clusters suggested by the elbow method appears to be 4, however, a better choice, more in line with our business intuition, seems to be 6 (where some clusters could be merged manually). Output for 6 clusters:



**Spectral Clustering**

This algorithm (run mostly on default settings with n\_clusters ranging from 2 to 10, assign\_label set to discretize, n\_jobs set to -1, and random\_state set to 7532) is not ideal (requires manual cluster merging). Sample output for 8 clusters:



**Affinity Propagation**

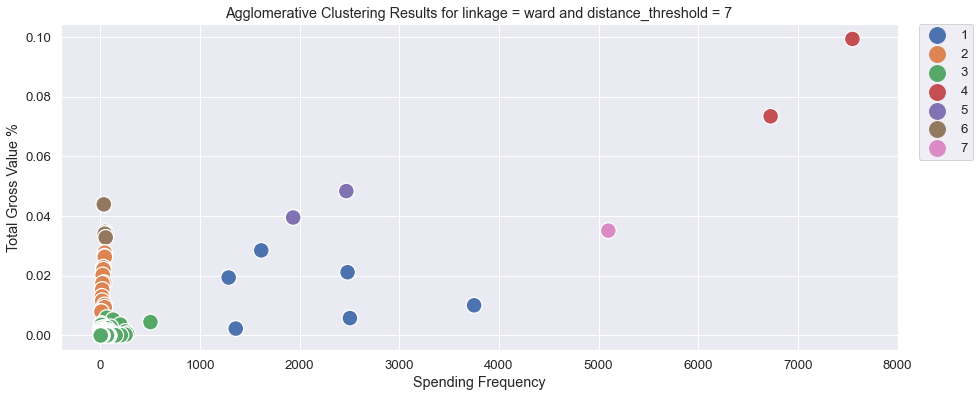
The results of this algorithm (run mostly on default settings with damping being either 0.7 or 0.75, max\_iter set to 1000 and random\_state set to 7532) are subpar as it either does not converge or produces too many (136) clusters.

**Mean Shift**

The results of this algorithm (run mostly on default settings with n\_jobs set to -1) are subpar as it produces too many (30) clusters which requires manual merging.

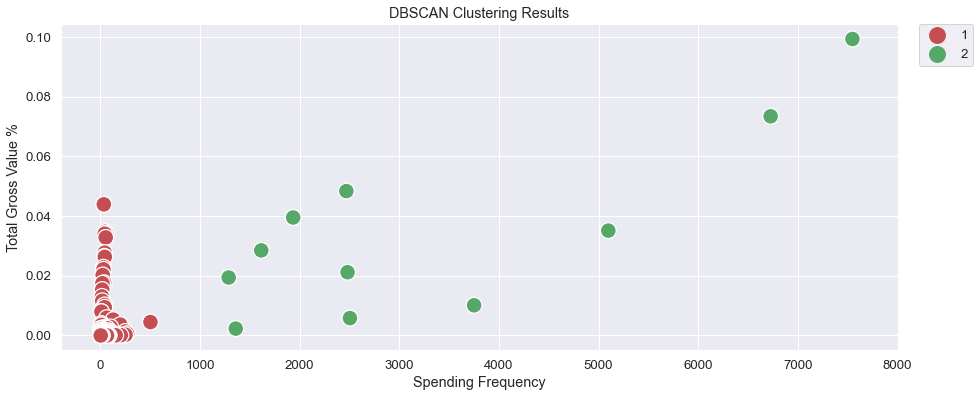
**Agglomerative Clustering**

This algorithm (run mostly on default settings with n\_clusters set to None, linkage assuming values from the list ['ward', 'complete', 'average', 'single'], and distance\_threshold assuming values from the list [3, 5, 7] is not ideal (requires manual cluster merging). Sample output (with 7 clusters) for linkage = ward and distance\_threshold = 7:



**DBSCAN**

This algorithm (run mostly on default settings with eps set to 2 and n\_jobs set to -1) appears to be best suited for our data. After applying this clustering algorithm we can see that the main cluster (composed of red points, see below) consists of low frequency products while those loosely scattered green points mostly represent the products exhibiting both relatively high value and high frequency.



And exactly those green points, listed below, form our list of potential saving opportunities (that are in line with our business intuition that we should look for such opportunities primarily among the high-value and high-moving products) which we will analyze presently.

|  |  |  | **Total Gross Value %** | **Spending Frequency** | **Spending Days** | **Cluster** |
| --- | --- | --- | --- | --- | --- | --- |
| **CoCd** | **Material** | **Short Text** |  |  |  |  |
| **7860** | **7101320.0** | **IB Ross Broiler Finisher Feed** | 0.099324 | 7547 | 89 | 2 |
| **IB Ross Broiler Starter Feed** | 0.073418 | 6726 | 89 | 2 |
| **9000** | **910970.0** | **Soya Bean - (MP)** | 0.048367 | 2468 | 88 | 2 |
| **910070.0** | **Soya Bean - (A)** | 0.039531 | 1934 | 87 | 2 |
| **7860** | **7101320.0** | **IB Ross Broiler Pre-Starter Feed** | 0.035124 | 5097 | 89 | 2 |
| **9000** | **910010.0** | **Maize** | 0.028506 | 1614 | 79 | 2 |
| **910000.0** | **Soya Bean** | 0.021192 | 2480 | 88 | 2 |
| **910860.0** | **Khandha** | 0.019418 | 1287 | 86 | 2 |
| **910420.0** | **Rice Bran Boiled** | 0.010110 | 3750 | 87 | 2 |
| **Rice Bran Raw** | 0.005824 | 2504 | 86 | 2 |
| **940730.0** | **Rice Husk (New)** | 0.002309 | 1358 | 84 | 2 |

Having selected these 11 candidates as potential savings opportunities we are going to employ the following strategy: Given the upward price trend, replace the net prices of a given product paid during the period with the median daily price of the first purchase day of that period in order to generate savings. Evidently, this strategy requires some form of verification (which we do by checking that actual savings are generated) and is based on the following assumptions:

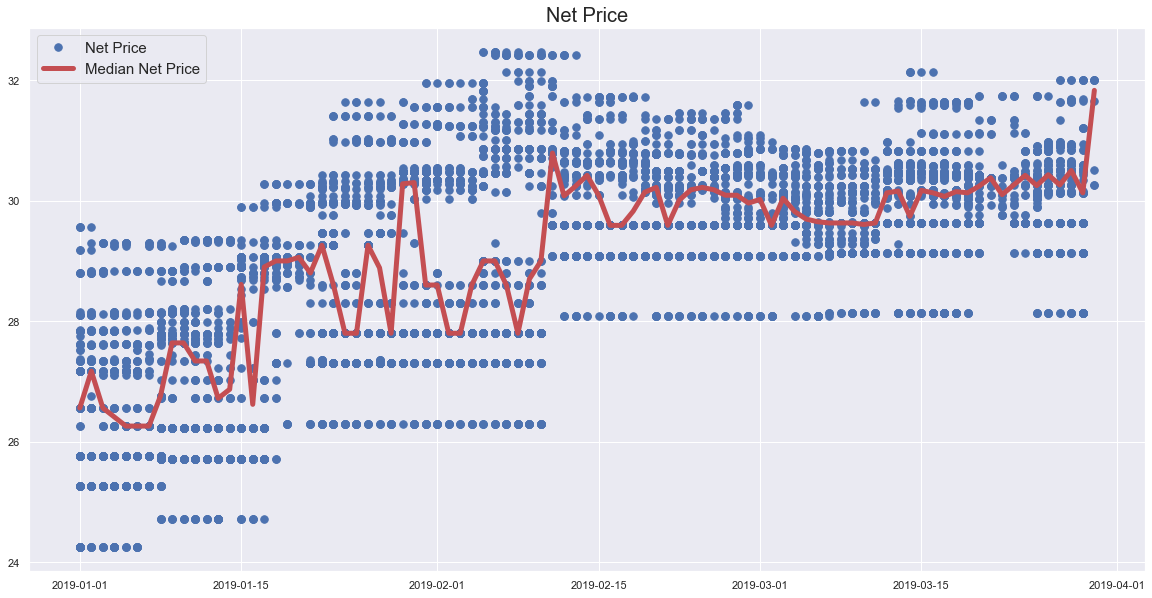
* The products are easy to store in large quantities for an extended period of time
* There is enough storage capacity for a bulk purchase (optionally, several deliveries for  one bulk order can be scheduled)

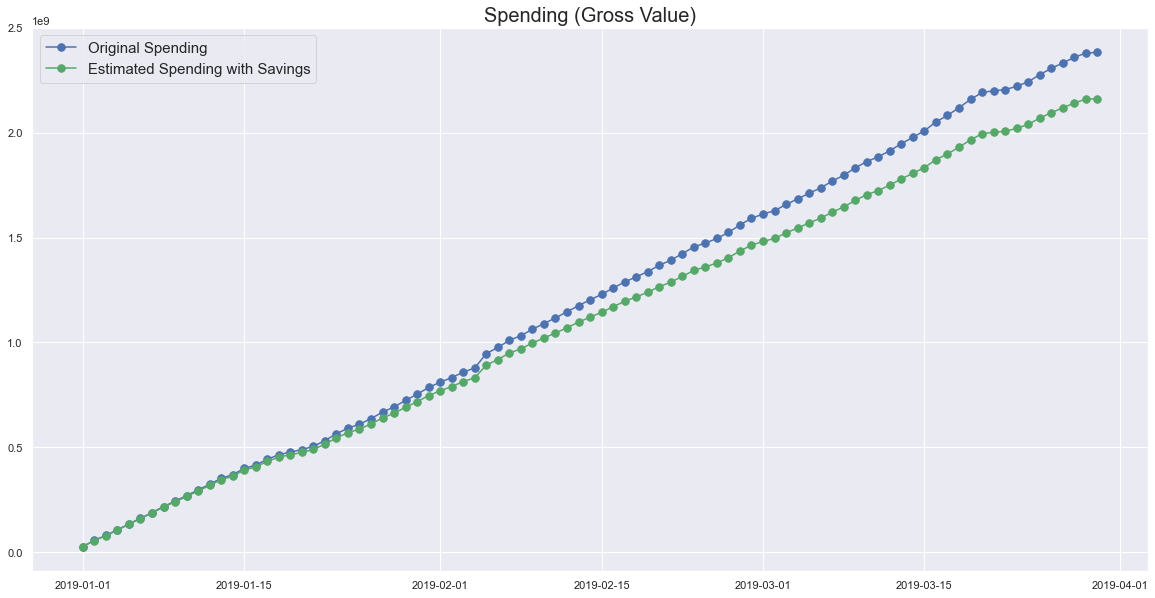
More details are as follows:

* For each product we select the appropriate company code, material, and short text combination to filter our 2019 data
* Then we compute the offset defined as the result of subtracting the net value from the gross value
* Next, we compute the daily median net price calculated over all available net prices per day whose purpose is twofold:
  + To indicate the "trend" of the net price in the given period
  + To serve as a reasonable (robust to outliers) estimate of the average daily net price
* Then we generate savings by calculating estimated spending (gross value) using the fixed median price from the first purchase day of the period as the price used in the whole period, multiplying this new price by the purchase order quantity to get the new net value and adding the offset to get the new gross value
* Finally, we verify that at the end of the period we actually get some savings

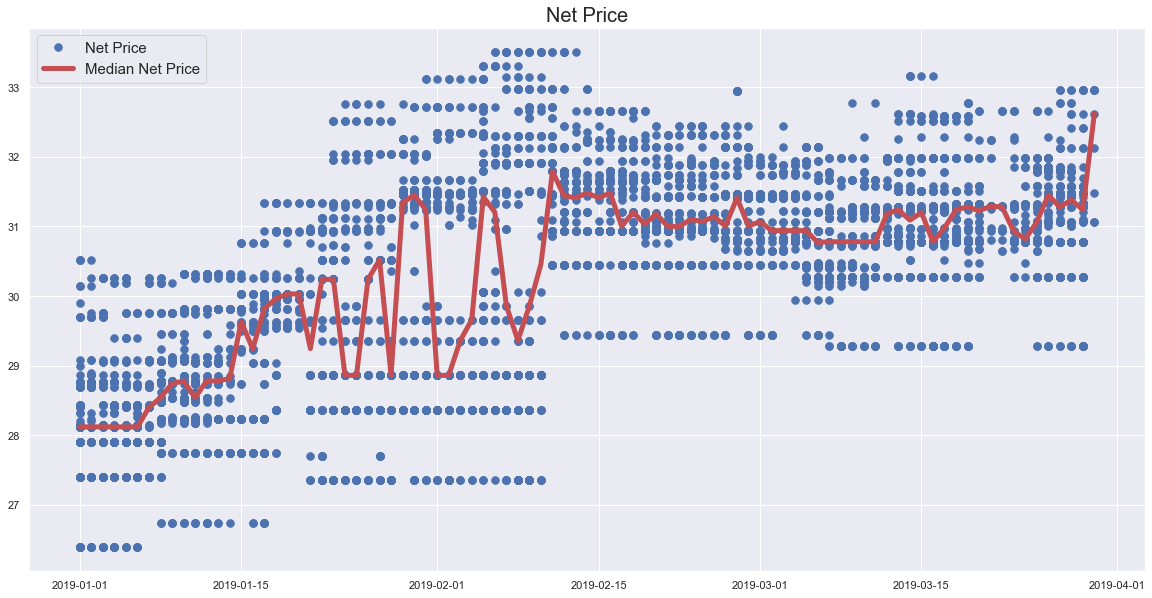
Below, for each of these 11 products, we present two charts that will help to visualize and validate our approach.

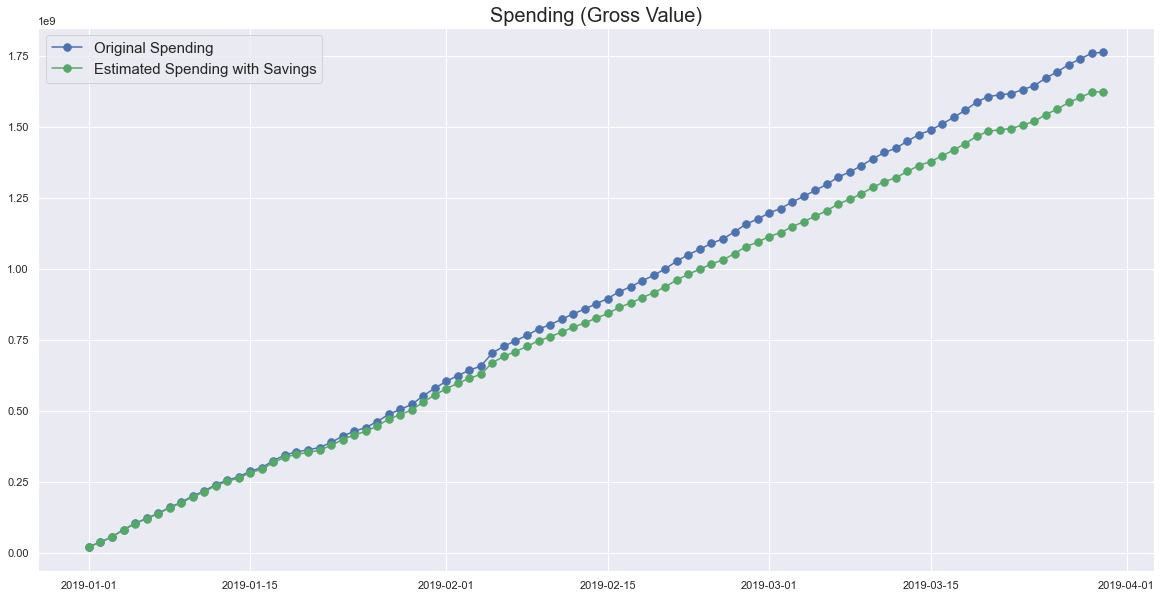
1. Company code: 7860, Material: 7101320, Description: IB Ross Broiler Finisher Feed



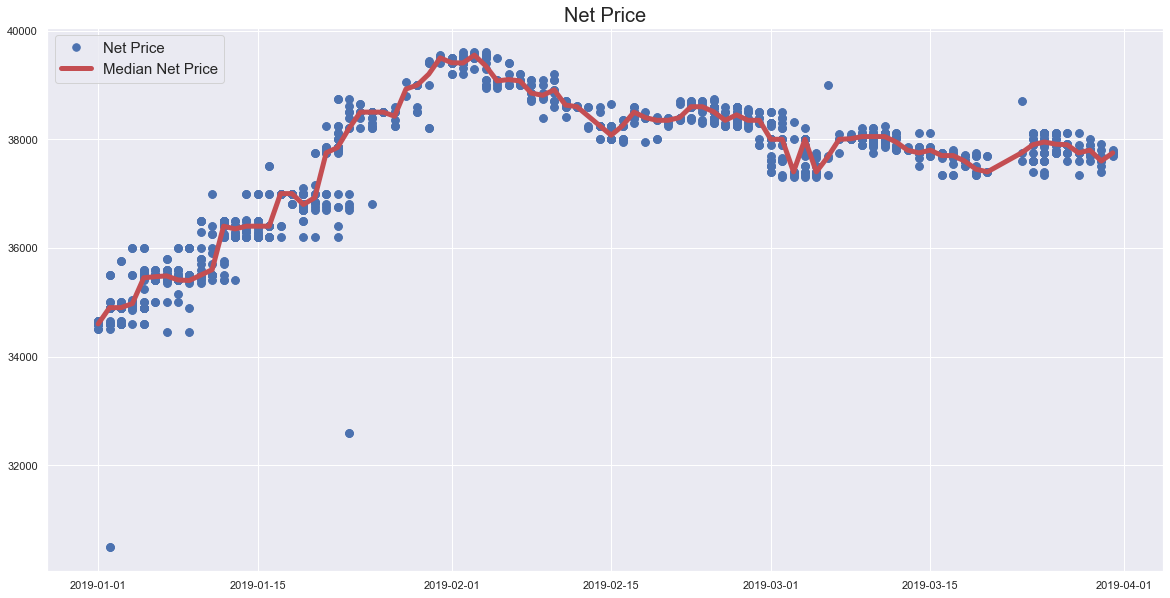


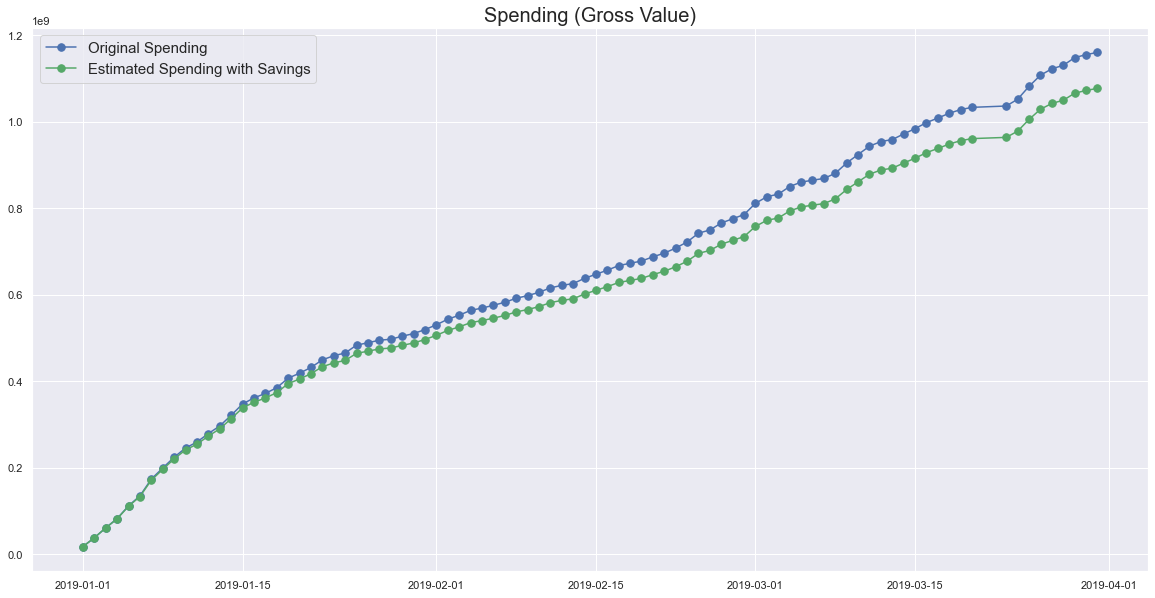
1. Company code: 7860, Material: 7101320, Description: IB Ross Broiler Starter Feed



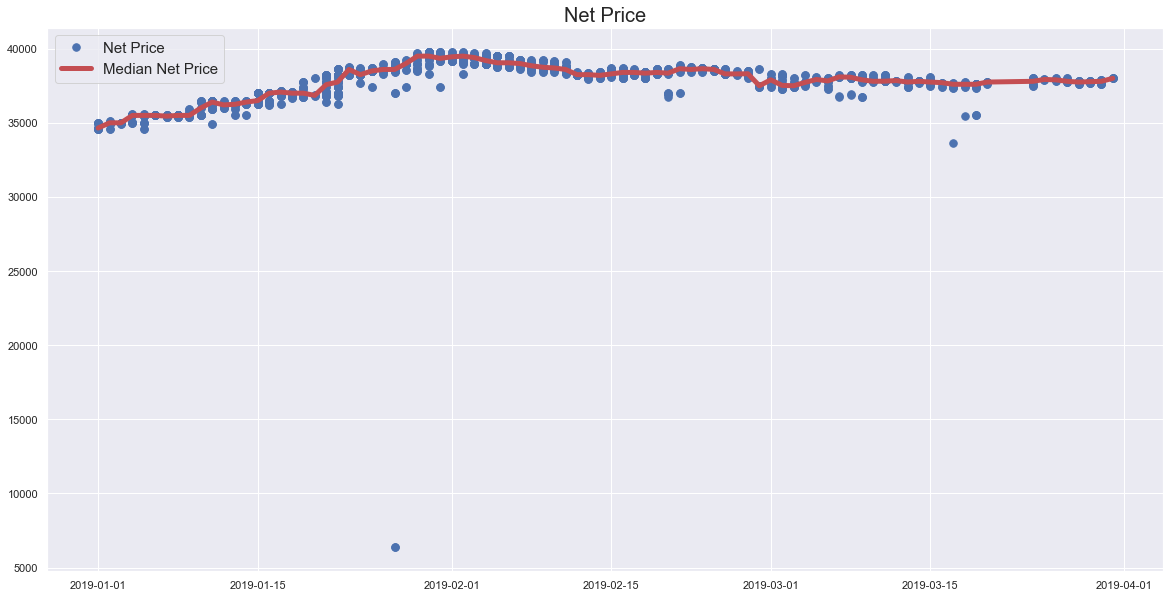


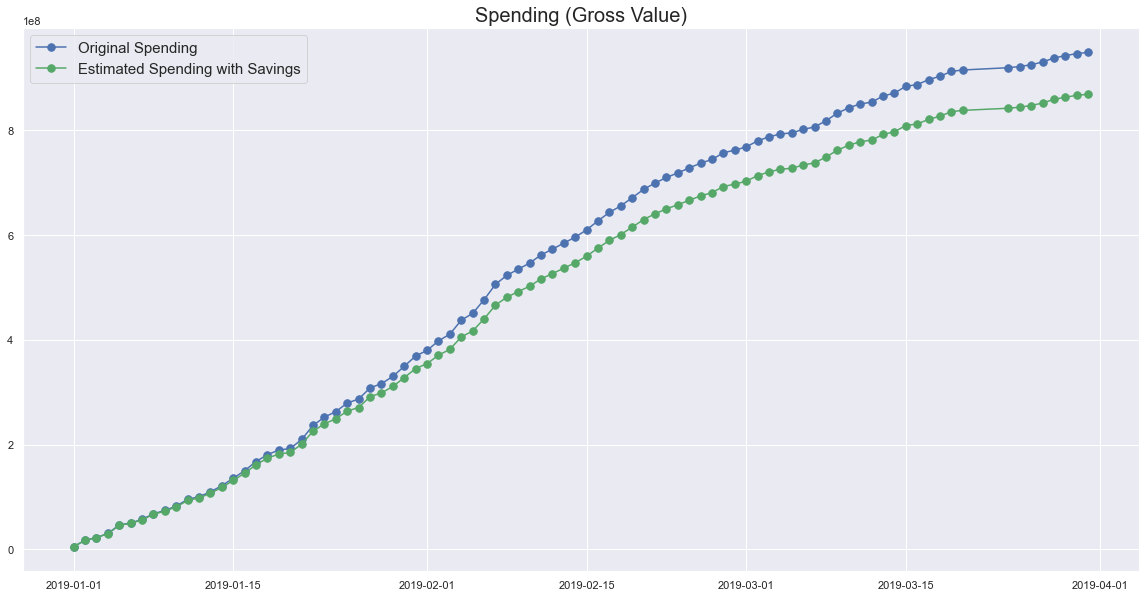
1. Company code: 9000, Material: 910970, Description: Soya Bean - (MP)



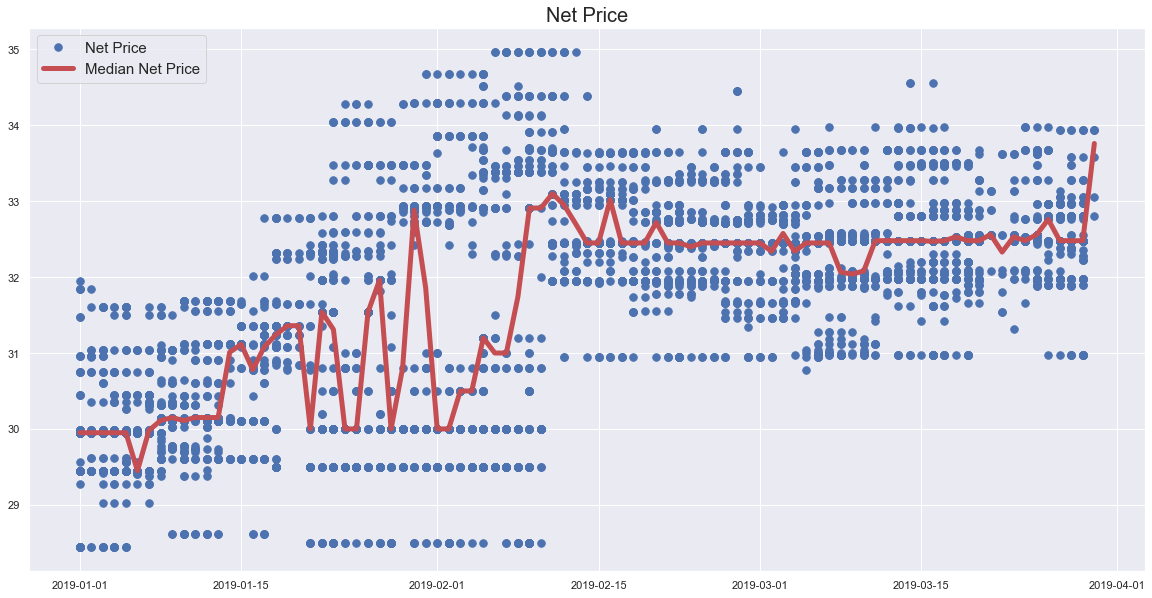


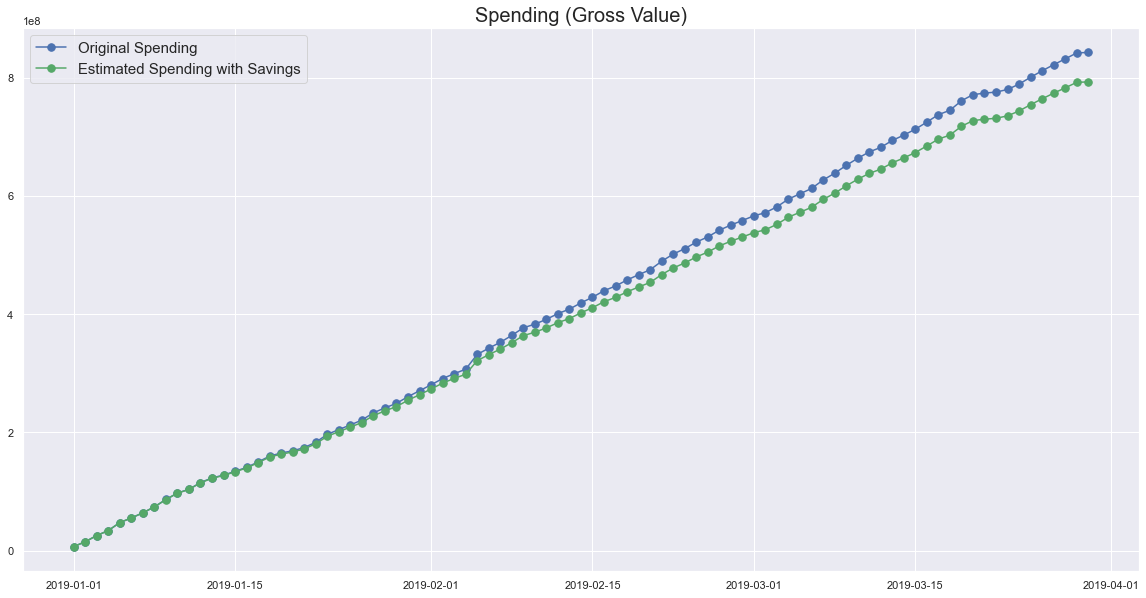
1. Company code: 9000, Material: 910070, Description: Soya Bean - (A)



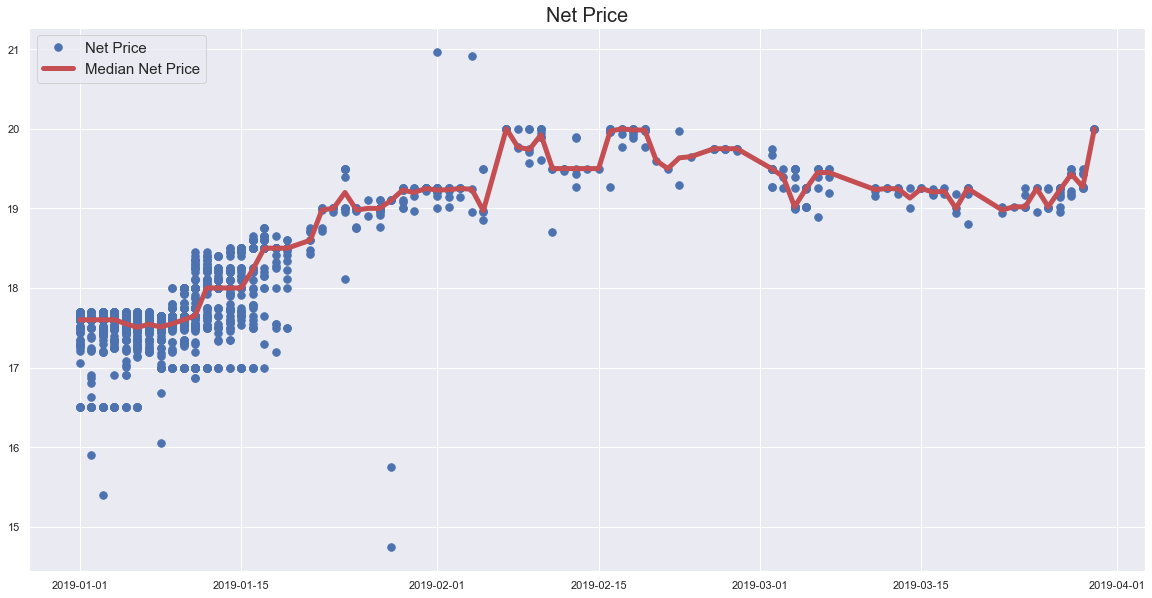


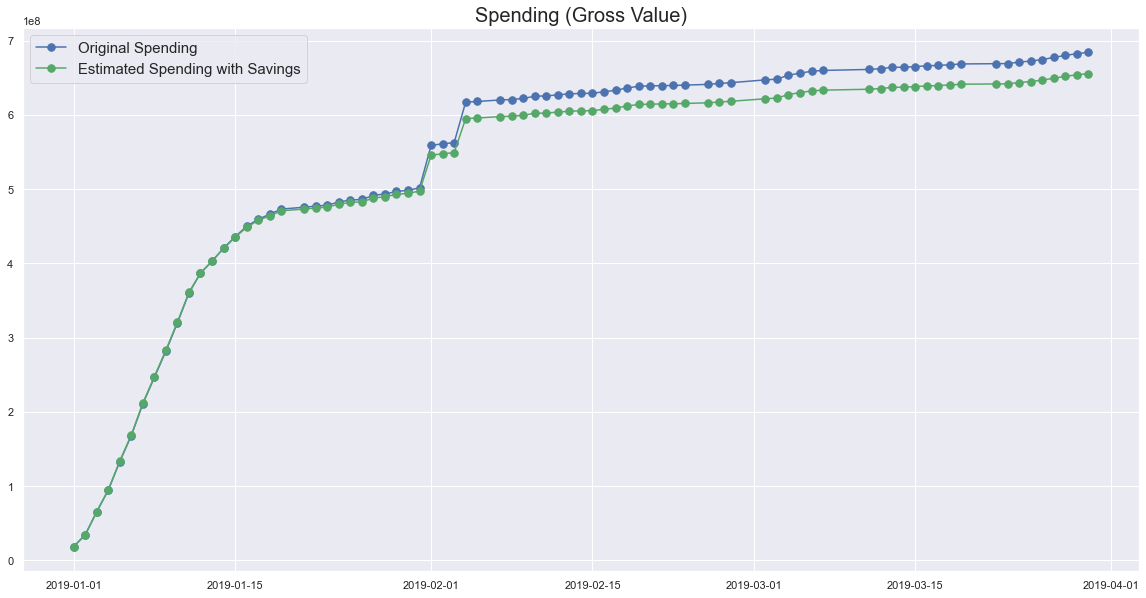
1. Company code: 7860, Material: 7101320, Description: IB Ross Broiler Pre-Starter Feed



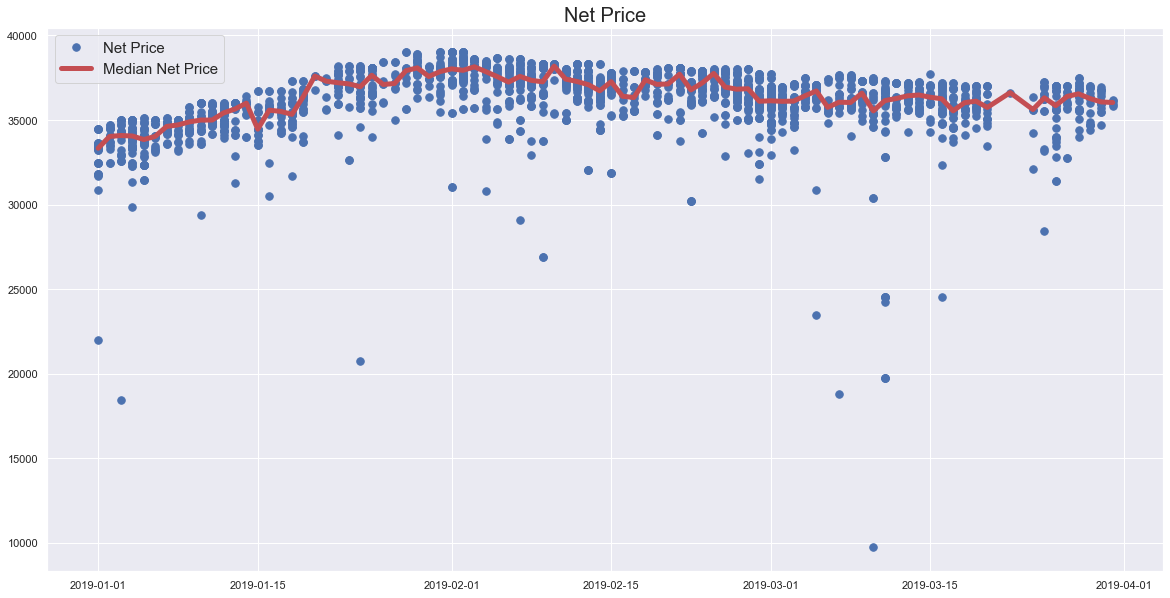


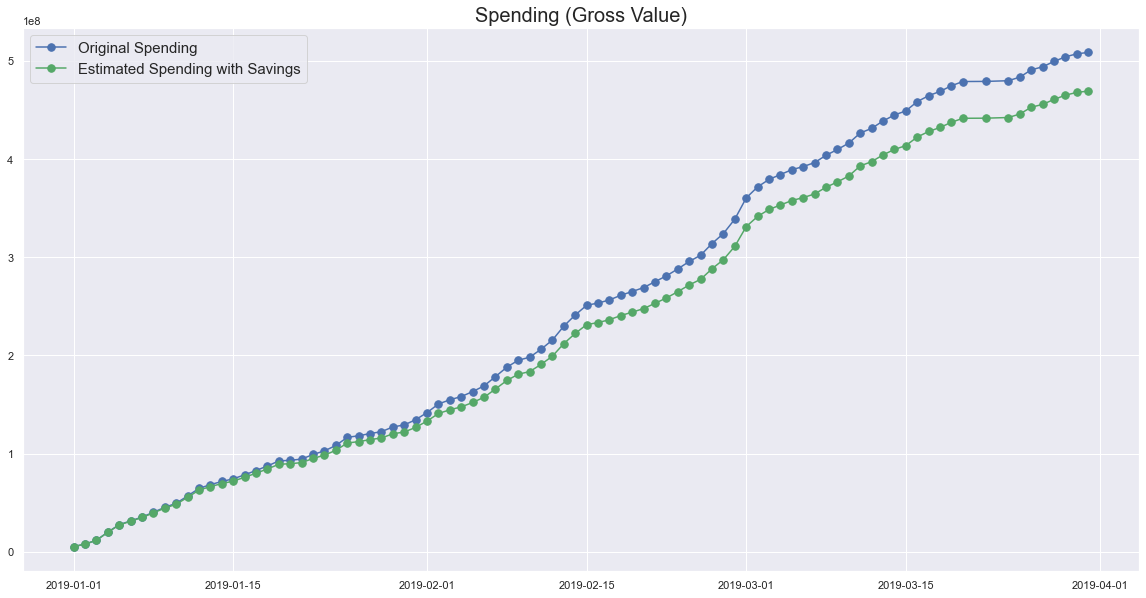
1. Company code: 9000, Material: 910010, Description: Maize



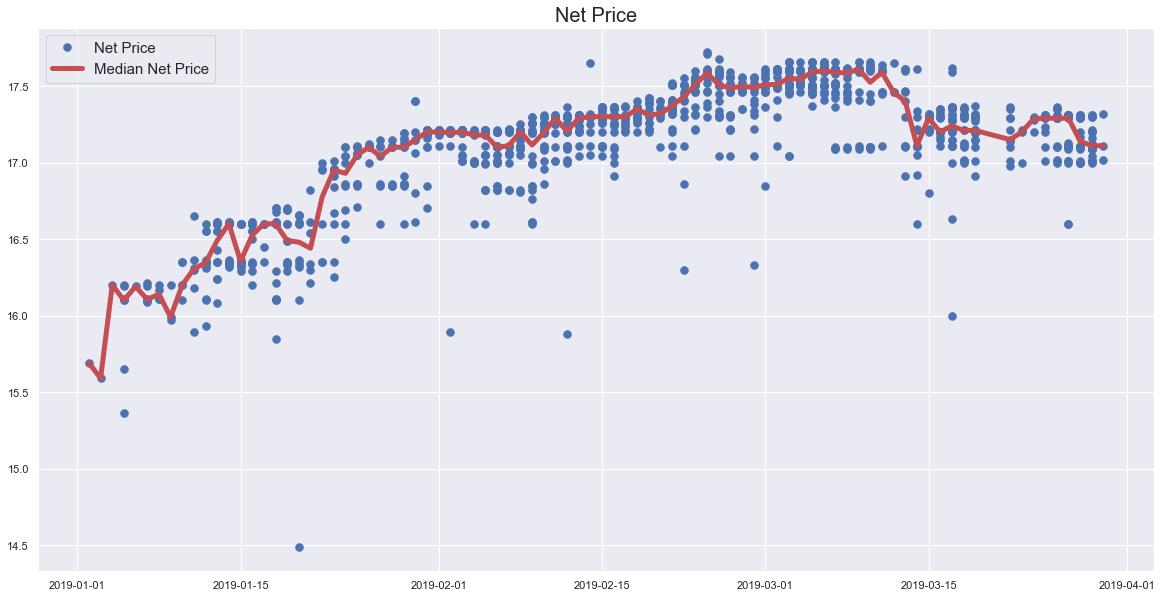


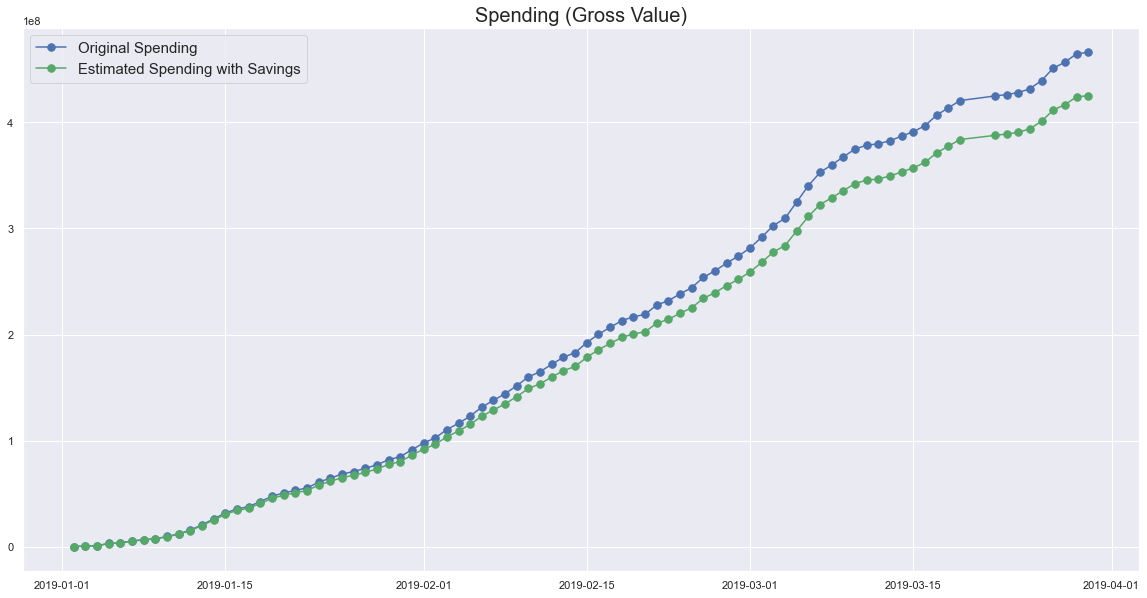
1. Company code: 9000, Material: 910000, Description: Soya Bean



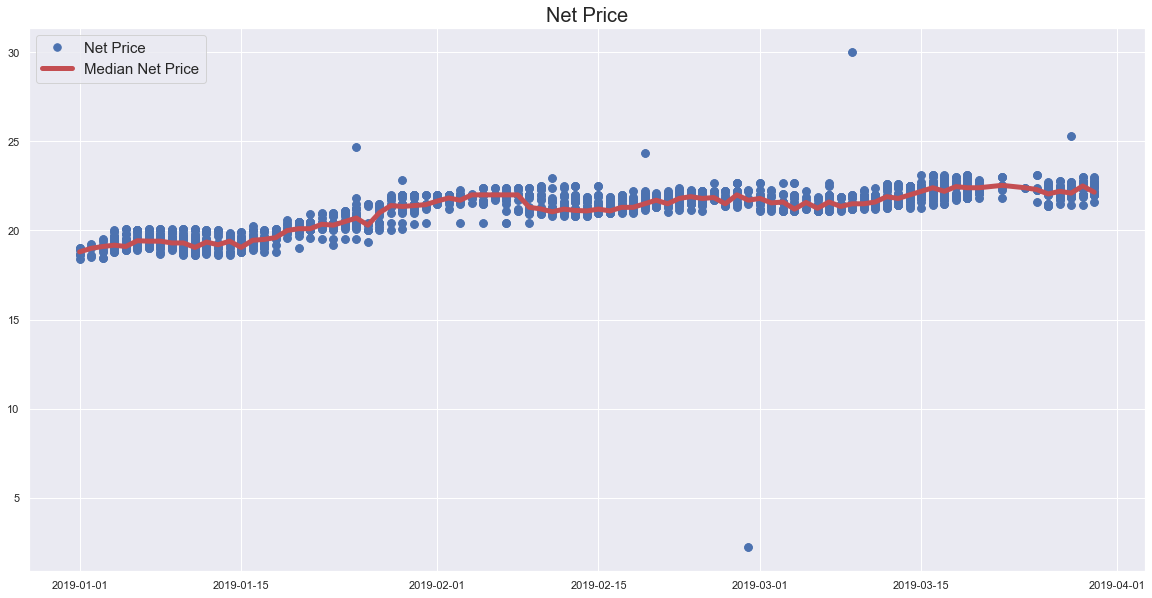


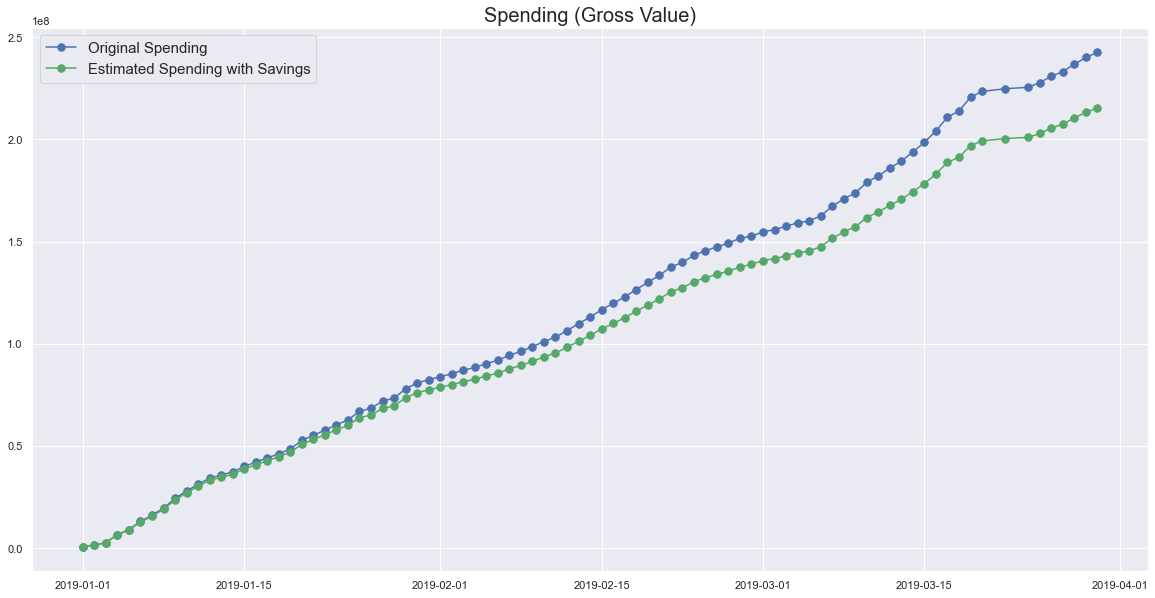
1. Company code: 9000, Material: 910860, Description: Khandha



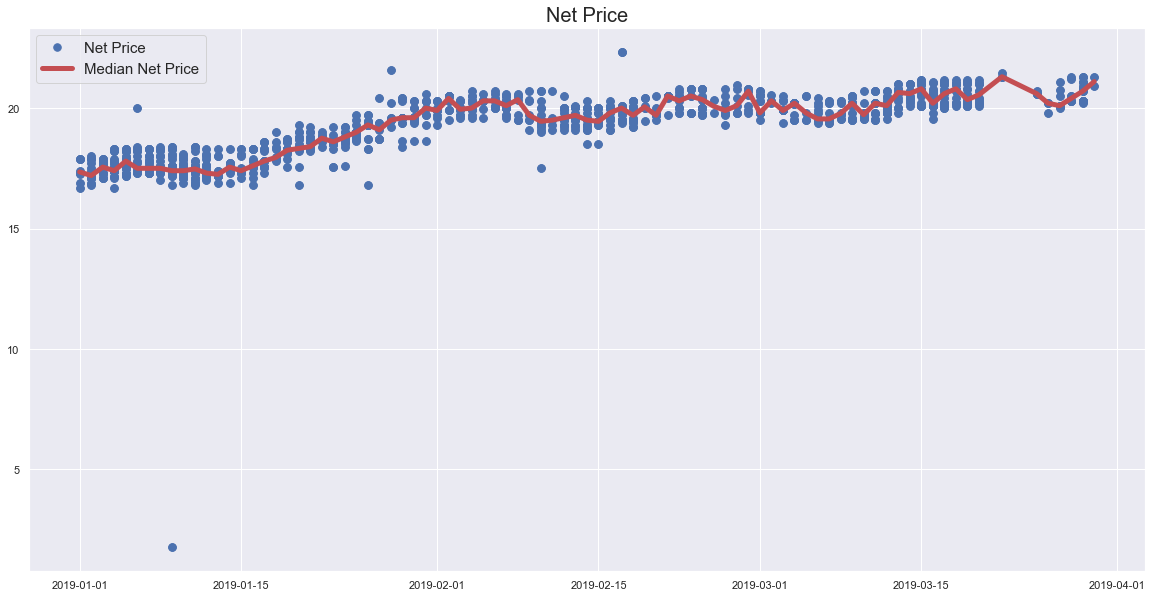


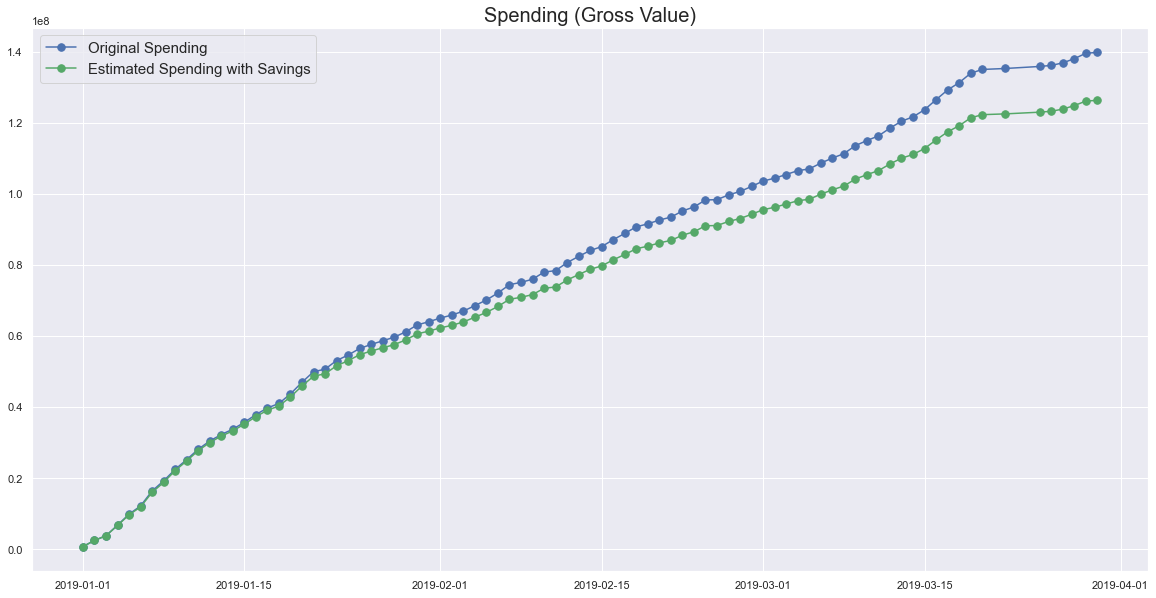
1. Company code: 9000, Material: 910420, Description: Rice Bran Boiled



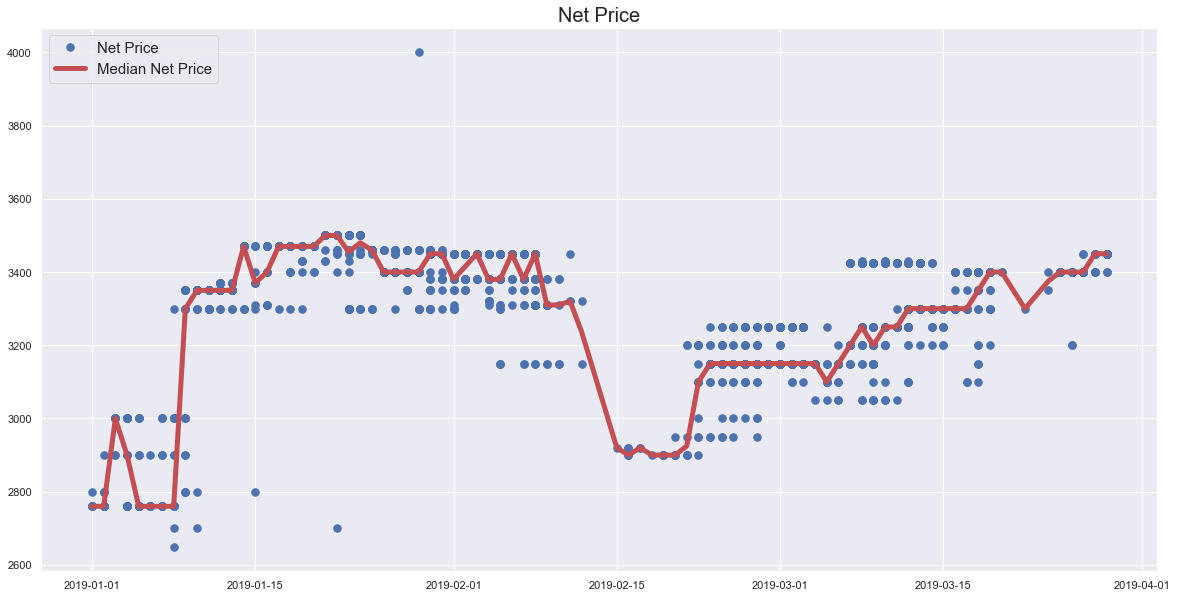


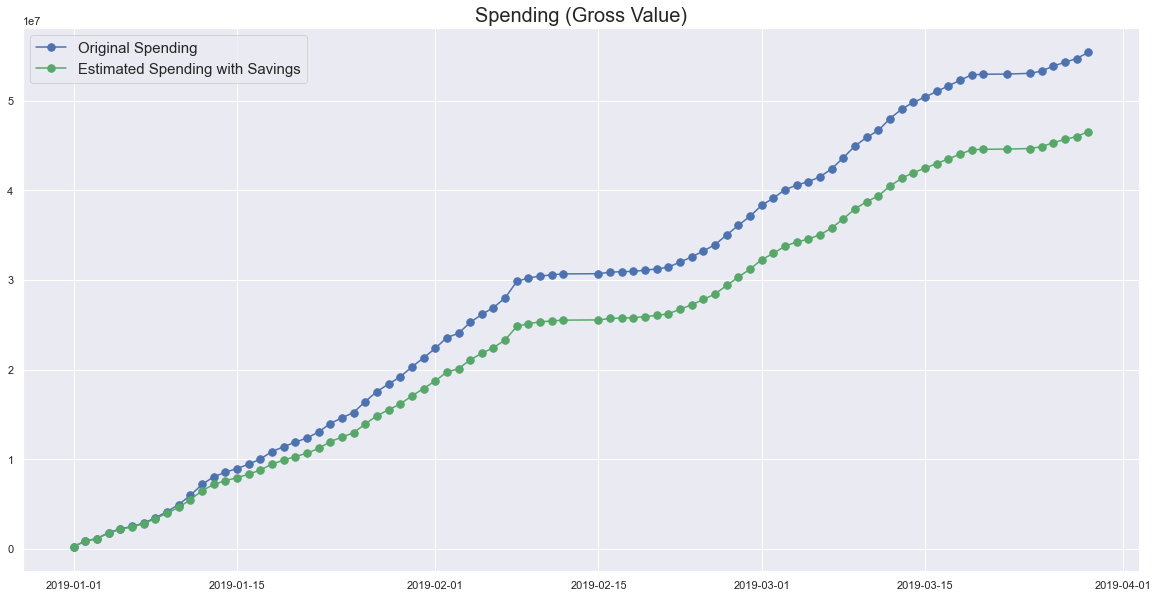
1. Company code: 9000, Material: 910420, Description: Rice Bran Raw





1. Company code: 9000, Material: 940730, Description: Rice Husk (New)





Finally, we compute the combined amount of savings:

Amount of savings: 729763363.47

Savings as the percentage of total spending: 3.041%

Savings as the percentage of spending in 2019: 3.255%

which is in the ballpark of 3% of total spending (in the entire original data set) and 3.25% of spending in 2019.

## Recommendation

Given the aforementioned assumptions and the fact that the data in 2019 span the first quarter only, we recommend that, in order to generate substantial savings, the following 11 products be bought at the beginning of the first quarter (as early as possible) for the entire quarter:

IB Ross Broiler Finisher Feed

IB Ross Broiler Starter Feed

IB Ross Broiler Pre-Starter Feed

Khandha

Maize

Rice Bran Boiled

Rice Bran Raw

Rice Husk (New)

Soya Bean

Soya Bean - (MP)

Soya Bean - (A)

# Discussion

In this section we are going to verify that for the data selected above for saving opportunities the records with a missing storage location are not significant enough to have an impact on the above findings.

We do so by showing the percentage of missing storage location data for such data records:

NaN percentage threshold: 0%

Number of columns with more than 0% of missing values: 1

Missing Data %

Columns with missing values

SLoc 0.04

which turns out to be just 0.04% and the percentage of total spending for such records:

Percentage of total spending for records with missing storage location: 0.111%

which turns out to be just 0.111%