Medical inventory optimization

Exploratory Data Analysis (SQL) by Rohit Paul

Software: MySQL Workbench

Business decisions based on the unclean 'projectfinaldata' table

1. Set the current database to "med_inventory".

```
USE med_inventory;
```

2. Displaying the table.

```
SELECT * FROM projectfinaldata LIMIT 20;
```

Output:

Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	ReturnQuantity	Final_Cost	Final_Sales	RtnMRP	Formulation	DrugName	SubCat	SubCat1
Sale	12018098765	Specialisation6	Department1	6-1-2022	1	0	55.406	59.26	0	Form1	ZINC ACETATE 20MG/5ML SYP	SYRUP & SUSPENSION	VITAMINS & MINERALS
Sale	12018103897	Specialisation7	Department1	7/23/2022	1	0	768.638	950.8	0	Form1	CEFTAZIDIME 2GM+AVIBACTAM 500MG	INJECTIONS	ANTI-INFECTIVES
Sale	12018101123	Specialisation2	Department3	6/23/2022	1	0	774.266	4004.214	0	Form2	EPTIFIBATIDE 0.75MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018079281	Specialisation40	Department1	3/17/2022	2	0	40.798	81.044	0	Form1	WATER FOR INJECTION 10ML SOLUTION	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018117928	Specialisation5	Department1	12/21/2022	1	0	40.434	40.504	0	Form1	LORAZEPAM 1MG	TABLETS & CAPSULES	CENTRAL NERVOUS SYSTEM
Return	12018103662	Specialisation2	Department1	7/15/2022	0	8	47.902	0	330	Form1	SALBUTAMOL 2.5MG	INHALERS & RESPULES	RESPIRATORY SYSTEM
Sale	12018097585	Specialisation2	Department1	5/22/2022	1	0	41.862	42.218	0	Form1	FUROSEMIDE 10MG/ML	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018077721	Specialisation4	Department1	1-12-2022	3	0	60.026	142.752	0	Form1	SODIUM CHLORIDE IVF 100ML	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018096500	Specialisation4	Department2	8/24/2022	2	0	49.856	94	0	Form2	SODIUM BICARBONATE 8.5% INJ	INJECTIONS	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018071649	Specialisation4	Department1	8/31/2022	1	0	258.86	319.8	0	Form1	PEPTIDE BASED DIET POWDER	NUTRITIONAL SUPPLEMENTS	NUTRITION
Sale	12018074894	Specialisation7	Department1	10-4-2022	3	0	114.592	290.4	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018088348	Specialisation4	Department1	4-2-2022	20	0	231.834	1294	0	Form1	N-ACETYLCYSTEINE 1000MG/5ML INJ	INJECTIONS	RESPIRATORY SYSTEM
Sale	12018101319	Specialisation16	Department2	7-1-2022	1	0	66.88	102.6	0	Form1	PROPOFOL 1% 20ML INJ	INJECTIONS	ANAESTHETICS
Sale	12018108547	Specialisation6	Department1	8/20/2022	8	0	52.204	343.84	0	Form1	PARACETAMOL 150MG	INJECTIONS	CENTRAL NERVOUS SYSTEM
Sale	12018080245	Specialisation7	Department1	7/29/2022	1	0	41.658	43.2	0	Form1	VITAMIN K 1ML INJ	INJECTIONS	CARDIOVASCULAR & HEMATOPOIETIC SYSTEM
Sale	12018115496	Specialisation2	Department1	11/26/2022	2	0	89.728	193.6	0	Form1	MULTIPLE ELECTROLYTES 500ML IVF	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018111286	Specialisation25	Department1	9-7-2022	1	0	49.352	60.8	0	Form1			
Sale	12018097033	Specialisation20	Department1	9/17/2022	2	0	40.34	81.1	0	Form1			
Return	12018122962	Specialisation54	Department1	12/19/2022	0	2	70.016	0	115	Form2	SODIUM CHLORIDE 0.9%	IV FLUIDS, ELECTROLYTES, TPN	INTRAVENOUS & OTHER STERILE SOLUTIONS
Sale	12018106746	Specialisation20	Department1	10/14/2022	2	0	40.34	81.1	0	Form1			

3. Calculating the first moment (measures of central tendency such as mean, median, mode) for the dataset.

Mean:

```
SELECT
  ROUND(AVG(Quantity), 2) AS mean_quantity,
  ROUND(AVG(ReturnQuantity), 2) AS mean_return_quantity,
  ROUND(AVG(Final_Cost), 2) AS mean_final_cost,
  ROUND(AVG(Final_Sales), 2) AS mean_final_sales,
  ROUND(AVG(RtnMRP), 2) AS mean_rtnmrp FROM projectfinaldata;
```

Output:

mean_quantity	mean_return_quantity	mean_final_cost	mean_final_sales	mean_rtnmrp
2.23	0.29	124.82	234.04	29.13

Median:

```
SELECT
  ROUND(AVG(Final_Cost), 2) AS median_final_cost,
  ROUND(AVG(Final_Sales), 2) AS median_final_sales,
  ROUND(AVG(Quantity), 2) AS median_quantity,
  ROUND(AVG(ReturnQuantity), 2) AS median_return_quantity,
  ROUND(AVG(RtnMRP), 2) AS median_rtnmrp
FROM (
  SELECT Final_Cost, Final_Sales, Quantity, ReturnQuantity, RtnMRP,
  ROW_NUMBER() OVER (ORDER BY Final_Cost) AS row_num,
  COUNT(*) OVER () AS total_rows FROM projectfinaldata
) AS subquery
WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEILING((total_rows + 1) / 2));
```

median_final_cost	median_final_sales	median_quantity	median_return_quantity	median_rtnmrp
53.65	83.44	1.00	0.00	0.00

Mode:

```
SELECT
  mode_quantity.mode_value AS mode_quantity,
  mode_return_quantity.mode_value AS mode_return_quantity,
  mode_final_cost.mode_value AS mode_final_cost,
  mode_final_sales.mode_value AS mode_final_sales,
  mode_rtnmrp.mode_value AS mode_rtnmrp
FROM (
  SELECT Quantity AS mode_value, COUNT(*) AS mode_count
  FROM projectfinaldata
  GROUP BY Quantity
  ORDER BY COUNT(*) DESC
  LIMIT 1
```

```
) AS mode_quantity,
(
  SELECT ReturnQuantity AS mode_value, COUNT(*) AS mode_count
  FROM projectfinaldata
  GROUP BY ReturnQuantity
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_return_quantity,
(
  SELECT Final_Cost AS mode_value, COUNT(*) AS mode_count
  FROM projectfinaldata
  GROUP BY Final_Cost
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_final_cost,
(
  SELECT Final_Sales AS mode_value, COUNT(*) AS mode_count
  FROM projectfinaldata
  GROUP BY Final_Sales
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_final_sales,
(
  SELECT RtnMRP AS mode_value, COUNT(*) AS mode_count
  FROM projectfinaldata
  GROUP BY RtnMRP
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_rtnmrp;
```

Output:

mode_quantity	mode_return_quantity	mode_final_cost	mode_final_sales	mode_rtnmrp
1	0	49.352	0	0

4. Calculating the second moment (measures of dispersion such as variance, standard deviation, range) for the dataset.

Variance:

```
SELECT
  ROUND(VARIANCE(Quantity), 2) AS variance_quantity,
  ROUND(VARIANCE(ReturnQuantity), 2) AS variance_return_quantity,
  ROUND(VARIANCE(Final_Cost), 2) AS variance_final_cost,
  ROUND(VARIANCE(Final_Sales), 2) AS variance_final_sales,
  ROUND(VARIANCE(RtnMRP), 2) AS variance_rtnmrp
FROM projectfinaldata;
```

Output:

variance_quantity	variance_return_quantity	variance_final_cost	variance_final_sales	variance_rtnmrp
26.34	2.7	216007.85	450560.41	33218.35

Standard Deviation:

```
SELECT
  ROUND(STDDEV(Quantity), 2) AS stddev_quantity,
  ROUND(STDDEV(ReturnQuantity), 2) AS stddev_return_quantity,
  ROUND(STDDEV(Final_Cost), 2) AS stddev_final_cost,
  ROUND(STDDEV(Final_Sales), 2) AS stddev_final_sales,
  ROUND(STDDEV(RtnMRP), 2) AS stddev_rtnmrp
FROM projectfinaldata;
```

Output:

stddev_quantity	stddev_return_quantity	stddev_final_cost	stddev_final_sales	stddev_rtnmrp
5.13	1.64	464.77	671.24	182.26

Range:

```
SELECT
  MAX(Quantity) - MIN(Quantity) AS range_quantity,
  MAX(ReturnQuantity) - MIN(ReturnQuantity) AS range_return_quantity,
  MAX(Final_Cost) - MIN(Final_Cost) AS range_final_cost,
  MAX(Final_Sales) - MIN(Final_Sales) AS range_final_sales,
  MAX(RtnMRP) - MIN(RtnMRP) AS range_rtnmrp
FROM projectfinaldata;
```

Output:

range_quantity	range_return_quantity	range_final_cost	range_final_sales	range_rtnmrp
150	50	33138	39490	8014

5. Calculating the third moment (skewness) for the dataset.

Skewness:

```
SELECT 'Quantity' AS column_name,
ROUND((SUM(POW(Quantity - (SELECT AVG(Quantity) FROM projectfinaldata), 3)) / (COUNT(*) * POW(STDDEV(Quantity), 3))), 2) AS skewness_value
FROM projectfinaldata
UNION ALL
SELECT 'ReturnQuantity' AS column_name,
ROUND((SUM(POW(ReturnQuantity - (SELECT AVG(ReturnQuantity) FROM projectfinaldata), 3)) / (COUNT(*) * POW(STDDEV(ReturnQuantity), 3))), 2) AS skewness_value
FROM projectfinaldata
UNION ALL
SELECT 'Final_Cost' AS column_name,
ROUND((SUM(POW(Final_Cost - (SELECT AVG(Final_Cost) FROM projectfinaldata), 3)) / (COUNT(*) * POW(STDDEV(Final_Cost), 3))), 2) AS skewness_value
FROM projectfinaldata
UNION ALL
SELECT 'Final_Sales' AS column_name,
```

```
ROUND((SUM(POW(Final_Sales - (SELECT AVG(Final_Sales) FROM projectfinaldata), 3)) / (COUNT(*) *
POW(STDDEV(Final_Sales), 3))), 2) AS skewness_value

FROM projectfinaldata

UNION ALL

SELECT 'RtnMRP' AS column_name,

ROUND((SUM(POW(RtnMRP - (SELECT AVG(RtnMRP) FROM projectfinaldata), 3)) / (COUNT(*) *
POW(STDDEV(RtnMRP), 3))), 2) AS skewness_value

FROM projectfinaldata;
```

Output:

column_name	skewness_value
Quantity	11.34
ReturnQuantity	17.17
Final_Cost	34.5
Final_Sales	21
RtnMRP	15.8

6. Calculating the fourth moment (kurtosis) for the dataset.

Kurtosis:

```
SELECT

ROUND((SUM(POWER(Quantity - avg_value, 4)) / (COUNT(Quantity) * POWER(STDDEV(Quantity),
4))), 2) AS kurtosis_quantity,

ROUND((SUM(POWER(ReturnQuantity - avg_value, 4)) / (COUNT(ReturnQuantity) *
POWER(STDDEV(ReturnQuantity), 4))), 2) AS kurtosis_return_quantity,

ROUND((SUM(POWER(Final_Cost - avg_value, 4)) / (COUNT(Final_Cost) *
POWER(STDDEV(Final_Cost), 4))), 2) AS kurtosis_final_cost,

ROUND((SUM(POWER(Final_Sales - avg_value, 4)) / (COUNT(Final_Sales) *
POWER(STDDEV(Final_Sales), 4))), 2) AS kurtosis_final_sales,

ROUND((SUM(POWER(RtnMRP - avg_value, 4)) / (COUNT(RtnMRP) * POWER(STDDEV(RtnMRP), 4))),
2) AS kurtosis_rtnmrp

FROM

 (SELECT

   AVG(Quantity) AS avg_value,

   STDDEV(Quantity) AS stddev_value,

   COUNT(Quantity) AS count_value
```

FROM projectfinaldata) AS subquery, projectfinaldata;

Output:

kurtosis_quantity	kurtosis_return_quantity	kurtosis_final_cost	kurtosis_final_sales	kurtosis_rtnmrp
183.09	341.5	2064.98	980.93	415.82

Business decisions based on 'cleaned_table' with pre-processed data

1. Calculating the first moment (measures of central tendency such as mean, median, mode) for the dataset.

Mean:

```
SELECT
  ROUND(AVG(Quantity), 2) AS mean_quantity,
  ROUND(AVG(ReturnQuantity), 2) AS mean_return_quantity,
  ROUND(AVG(Final_Cost), 2) AS mean_final_cost,
  ROUND(AVG(Final_Sales), 2) AS mean_final_sales,
  ROUND(AVG(RtnMRP), 2) AS mean_rtnmrp FROM cleaned_table;
```

Output:

mean_quantity	mean_return_quantity	mean_final_cost	mean_final_sales	mean_rtnmrp
1.78	0.17	89.9	167.78	12.52

Median:

```
SELECT
  ROUND(AVG(Final_Cost), 2) AS median_final_cost,
  ROUND(AVG(Final_Sales), 2) AS median_final_sales,
  ROUND(AVG(Quantity), 2) AS median_quantity,
  ROUND(AVG(ReturnQuantity), 2) AS median_return_quantity,
  ROUND(AVG(RtnMRP), 2) AS median_rtnmrp
FROM (
  SELECT Final_Cost, Final_Sales, Quantity, ReturnQuantity, RtnMRP,
  ROW_NUMBER() OVER (ORDER BY Final_Cost) AS row_num,
  COUNT(*) OVER () AS total_rows FROM cleaned_table
```

```
) AS subquery
WHERE row_num IN (FLOOR((total_rows + 1) / 2), CEILING((total_rows + 1) / 2));
```

median_final_cost	median_final_sales	median_quantity	median_return_quantity	median_rtnmrp
52.99	56.44	1.00	0.00	0.00

Mode:

```
SELECT
  mode_quantity.mode_value AS mode_quantity,
  mode_return_quantity.mode_value AS mode_return_quantity,
  mode_final_cost.mode_value AS mode_final_cost,
  mode_final_sales.mode_value AS mode_final_sales,
  mode_rtnmrp.mode_value AS mode_rtnmrp
FROM (
  SELECT Quantity AS mode_value, COUNT(*) AS mode_count
  FROM cleaned_table
  GROUP BY Quantity
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_quantity,
(
  SELECT ReturnQuantity AS mode_value, COUNT(*) AS mode_count
  FROM cleaned_table
  GROUP BY ReturnQuantity
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_return_quantity,
(
  SELECT Final_Cost AS mode_value, COUNT(*) AS mode_count
  FROM cleaned_table
  GROUP BY Final_Cost
```

```
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_final_cost,
(
  SELECT Final_Sales AS mode_value, COUNT(*) AS mode_count
  FROM cleaned_table
  GROUP BY Final_Sales
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_final_sales,
(
  SELECT RtnMRP AS mode_value, COUNT(*) AS mode_count
  FROM cleaned_table
  GROUP BY RtnMRP
  ORDER BY COUNT(*) DESC
  LIMIT 1
) AS mode_rtnmrp;
```

Output:

mode_quantity	mode_return_quantity	mode_final_cost	mode_final_sales	mode_rtnmrp
1	0	49.352	0	0

2. Calculating the second moment (measures of dispersion such as variance, standard deviation, range) for the dataset.

Variance:

```
SELECT
  ROUND(VARIANCE(Quantity), 2) AS variance_quantity,
  ROUND(VARIANCE(ReturnQuantity), 2) AS variance_return_quantity,
  ROUND(VARIANCE(Final_Cost), 2) AS variance_final_cost,
  ROUND(VARIANCE(Final_Sales), 2) AS variance_final_sales,
  ROUND(VARIANCE(RtnMRP), 2) AS variance_rtnmrp
```

FROM cleaned_table;

Output:

variance_quantity	variance_return_quantity	variance_final_cost	variance_final_sales	variance_rtnmrp
3.21	0.32	18874.16	74433.94	2176.52

Standard Deviation:

```
SELECT
  ROUND(STDDEV(Quantity), 2) AS stddev_quantity,
  ROUND(STDDEV(ReturnQuantity), 2) AS stddev_return_quantity,
  ROUND(STDDEV(Final_Cost), 2) AS stddev_final_cost,
  ROUND(STDDEV(Final_Sales), 2) AS stddev_final_sales,
  ROUND(STDDEV(RtnMRP), 2) AS stddev_rtnmrp
FROM cleaned_table;
```

Output:

stddev_quantity	stddev_return_quantity	stddev_final_cost	stddev_final_sales	stddev_rtnmrp
1.79	0.56	137.38	272.83	46.65

Range:

```
SELECT
  MAX(Quantity) - MIN(Quantity) AS range_quantity,
  MAX(ReturnQuantity) - MIN(ReturnQuantity) AS range_return_quantity,
  MAX(Final_Cost) - MIN(Final_Cost) AS range_final_cost,
  MAX(Final_Sales) - MIN(Final_Sales) AS range_final_sales,
  MAX(RtnMRP) - MIN(RtnMRP) AS range_rtnmrp
FROM cleaned_table;
```

Output:

range_quantity	range_return_quantity	range_final_cost	range_final_sales	range_rtnmrp
17	4	1363.11	2143.64	555

3. Calculating the third moment (skewness) for the dataset.

Skewness:

```
SELECT 'Quantity' AS column_name,

ROUND((SUM(POW(Quantity - (SELECT AVG(Quantity) FROM cleaned_table), 3)) / (COUNT(*) * POW(STDDEV(Quantity), 3))), 2) AS skewness_value

FROM cleaned_table

UNION ALL

SELECT 'ReturnQuantity' AS column_name,

ROUND((SUM(POW(ReturnQuantity - (SELECT AVG(ReturnQuantity) FROM cleaned_table), 3)) / (COUNT(*) * POW(STDDEV(ReturnQuantity), 3))), 2) AS skewness_value

FROM cleaned_table

UNION ALL

SELECT 'Final_Cost' AS column_name,

ROUND((SUM(POW(Final_Cost - (SELECT AVG(Final_Cost) FROM cleaned_table), 3)) / (COUNT(*) * POW(STDDEV(Final_Cost), 3))), 2) AS skewness_value

FROM cleaned_table

UNION ALL

SELECT 'Final_Sales' AS column_name,

ROUND((SUM(POW(Final_Sales - (SELECT AVG(Final_Sales) FROM cleaned_table), 3)) / (COUNT(*) * POW(STDDEV(Final_Sales), 3))), 2) AS skewness_value

FROM cleaned_table

UNION ALL

SELECT 'RtnMRP' AS column_name,

ROUND((SUM(POW(RtnMRP - (SELECT AVG(RtnMRP) FROM cleaned_table), 3)) / (COUNT(*) * POW(STDDEV(RtnMRP), 3))), 2) AS skewness_value

FROM cleaned_table;
```

Output:

column_name	skewness_value
Quantity	2.95
ReturnQuantity	4.02
Final_Cost	5.69
Final_Sales	4.25
RtnMRP	5.31

4. Calculating the fourth moment (kurtosis) for the dataset.

Kurtosis:

```
SELECT

ROUND((SUM(POWER(Quantity - avg_value, 4)) / (COUNT(Quantity) * POWER(STDDEV(Quantity), 4))), 2) AS kurtosis_quantity,

ROUND((SUM(POWER(ReturnQuantity - avg_value, 4)) / (COUNT(ReturnQuantity) * POWER(STDDEV(ReturnQuantity), 4))), 2) AS kurtosis_return_quantity,

ROUND((SUM(POWER(Final_Cost - avg_value, 4)) / (COUNT(Final_Cost) * POWER(STDDEV(Final_Cost), 4))), 2) AS kurtosis_final_cost,

ROUND((SUM(POWER(Final_Sales - avg_value, 4)) / (COUNT(Final_Sales) * POWER(STDDEV(Final_Sales), 4))), 2) AS kurtosis_final_sales,

ROUND((SUM(POWER(RtnMRP - avg_value, 4)) / (COUNT(RtnMRP) * POWER(STDDEV(RtnMRP), 4))), 2) AS kurtosis_rtnmrp

FROM

 (SELECT

   AVG(Quantity) AS avg_value,

   STDDEV(Quantity) AS stddev_value,

   COUNT(Quantity) AS count_value

 FROM projectfinaldata) AS subquery, cleaned_table;
```

Output:

kurtosis_quantity	kurtosis_return_quantity	kurtosis_final_cost	kurtosis_final_sales	kurtosis_rtnmrp
12.54	219.98	56.19	36.63	43.03

Comparison table showing the business decisions results for the unclean and clean data:

Parameter	Unclean Data	Clean Data
Mean	Quantity: 2.23	Quantity: 1.78
	ReturnQuantity: 0.29	ReturnQuantity: 0.17
	Final_Cost: 124.82	Final_Cost: 89.9
	Final_Sales: 234.04	Final_Sales: 167.78
	RtnMRP: 29.13	RtnMRP: 12.52
Median	Quantity: 1	Quantity: 1
	ReturnQuantity: 0	ReturnQuantity: 0
	Final_Cost: 53.65	Final_Cost: 52.99
	Final_Sales: 83.44	Final_Sales: 56.44
	RtnMRP: 0	RtnMRP: 0
Mode	Quantity: 1	Quantity: 1
	ReturnQuantity: 0	ReturnQuantity: 0
	Final_Cost: 49.352	Final_Cost: 49.352
	Final_Sales: 0	Final_Sales: 0
	RtnMRP: 0	RtnMRP: 0
Variance	Quantity: 26.34	Quantity: 3.21
	ReturnQuantity: 2.7	ReturnQuantity: 0.32
	Final_Cost: 216007.85	Final_Cost: 18874.16
	Final_Sales: 450560.41	Final_Sales: 74433.94
	RtnMRP: 33218.35	RtnMRP: 2176.52
Standard Deviation	Quantity: 5.13	Quantity: 1.79
	ReturnQuantity: 1.64	ReturnQuantity: 0.56
	Final_Cost: 464.77	Final_Cost: 137.38
	Final_Sales: 671.24	Final_Sales: 272.83
	RtnMRP: 182.26	RtnMRP: 46.65
Range	Quantity: 150	Quantity: 17
	ReturnQuantity: 50	ReturnQuantity: 4
	Final_Cost: 33138	Final_Cost: 1363.11
	Final_Sales: 39490	Final_Sales: 2143.64
	RtnMRP: 8014	RtnMRP: 555
Skewness	Quantity: 11.34	Quantity: 2.95
	ReturnQuantity: 17.17	ReturnQuantity: 4.02
	Final_Cost: 34.5	Final_Cost: 5.69
	Final_Sales: 21	Final_Sales: 4.25
	RtnMRP: 15.8	RtnMRP: 5.31
Kurtosis	Quantity: 183.09	Quantity: 12.54
	ReturnQuantity: 341.5	ReturnQuantity: 219.98
	Final_Cost: 2064.98	Final_Cost: 56.19
	Final_Sales: 980.93	Final_Sales: 36.63
	RtnMRP: 415.82	RtnMRP: 43.03

Observation:

Overall, the results indicate that the unclean data exhibits higher mean, variance, standard deviation, range, skewness, and kurtosis values compared to the clean data. This suggests greater inconsistencies, variability, and potential outliers in the unclean data. Cleaning the data has resulted in more stable and normalized distributions with reduced variability and potential biases, making it more reliable for business decision-making.

Bounce Rate Analysis:

1. Finding the percentage of customers who bounced (returned a product with a final sale price of 0) out of the total number of customers.

```
SELECT ROUND((bounced_customers / total_customers) * 100, 2) AS bounce_rate
FROM
 (SELECT COUNT(DISTINCT Patient_ID) AS total_customers
 FROM cleaned_table
 WHERE Typeofsales IN ('Sale', 'Return')) AS t1,
 (SELECT COUNT(DISTINCT Patient_ID) AS bounced_customers
 FROM cleaned_table
 WHERE Typeofsales = 'Return' AND Final_Sales = 0) AS t2;
```

Output:

bounce_rate
22.85

Insight:

We can understand that around 22.85% of customers in the 'cleaned_table' faced a situation where they returned medicines with a Final_Sales value of 0. This means that a significant portion of customers did not get the medicines they needed, which could lead to dissatisfaction. To improve business success and increase revenue, it is important to reduce this bounce rate by ensuring customers receive the medicines they require.

2. Finding the number of drugs in each subcategory that have been returned without making a sale (Final_Sales = 0).

```
SELECT SubCat, COUNT(DISTINCT DrugName) AS num_returned_drugs
FROM cleaned_table
WHERE Typeofsales = 'Return' AND Final_Sales = 0
GROUP BY SubCat
ORDER BY num_returned_drugs DESC;
```

Output:

SubCat	num_returned_drugs
INJECTIONS	98
TABLETS & CAPSULES	63
IV FLUIDS, ELECTROLYTES, TPN	18
INHALERS & RESPULES	10
SYRUP & SUSPENSION	8
OINTMENTS, CREAMS & GELS	7
DROPS	6
POWDER	6
LIQUIDS & SOLUTIONS	5
PESSARIES & SUPPOSITORIES	4
NUTRITIONAL SUPPLEMENTS	3
VACCINE	2
LOTIONS	1
PATCH	1
unknown	1

Insight:

We can observe that the subcategory "INJECTIONS" has the highest count of returned drug names with 98 occurrences, followed by the subcategory "TABLETS & CAPSULES" with 63 occurrences, indicating a potential issue with customer satisfaction, product quality, or other factors that lead to returns for these two subcategories.

3. Finding the formulation with the highest return count within the "INJECTIONS" and "TABLETS & CAPSULES" subcategories.

```
SELECT SubCat, Formulation, return_count
FROM (
    SELECT SubCat, Formulation, COUNT(*) AS return_count,
        ROW_NUMBER() OVER (PARTITION BY SubCat ORDER BY COUNT(*) DESC) AS rn
    FROM cleaned_table
    WHERE Typeofsales = 'Return' AND Final_Sales = 0 AND SubCat IN ('INJECTIONS', 'TABLETS & CAPSULES')
    GROUP BY SubCat, Formulation
) AS subquery
WHERE rn = 1;
```

Output:

SubCat	Formulation	return_count
INJECTIONS	Form1	398
TABLETS & CAPSULES	Form1	77

Insight:

We can observe that within the "INJECTIONS" subcategory, the Formulation "Form1" has the highest return count with 398 occurrences. Similarly, within the "TABLETS & CAPSULES" subcategory, the Formulation "Form1" again has the highest return count with 77 occurrences.

4. Finding the count of occurrences of Formulation "Form1" for each Department (Dept) where the SubCat is either "INJECTIONS" or "TABLETS & CAPSULES".

```
SELECT Dept, COUNT(*) AS form1_count

FROM cleaned_table

WHERE Formulation = 'Form1' AND SubCat IN ('INJECTIONS', 'TABLETS & CAPSULES')

GROUP BY Dept

ORDER BY form1_count DESC;
```

Output:

Dept	form1_count
Department1	5226
Department2	782
Department3	99

Insight: The Formulation "Form1" appears to be significantly more prevalent in Department1 compared to other departments which likely corresponds to the Pharmacy department, has a high demand or usage of the "Form1" formulation. This could indicate that "Form1" is a commonly prescribed or requested medication within this department.

5. Finding the count of occurrences of Typeofsales as 'Return' for each Department (Dept).

```
SELECT Dept, COUNT(*) AS return_count

FROM cleaned_table

WHERE Typeofsales = 'Return'

GROUP BY Dept;
```

Output:

Dept	return_count
Department1	1405
Department2	1

Insight: Department1 has a relatively higher count of return occurrences compared to other departments. This suggests that there may be more instances of customers returning products in Department1.

6. Finding the count of occurrences of Typeofsales as 'Return' for each Specialisation within Department1 and Formulation as 'Form1'

```
SELECT Specialisation, COUNT(*) AS return_count
FROM cleaned_table
WHERE Typeofsales = 'Return' AND Dept = 'Department1' AND Formulation = 'Form1'
GROUP BY Specialisation
ORDER BY return_count DESC;
```

Output:

Specialisation	return_count
Specialisation4	197
Specialisation7	143
Specialisation5	40
Specialisation6	38
Specialisation11	37
Specialisation8	33
Specialisation3	32
Specialisation2	32
Specialisation1	29
Specialisation20	25
Specialisation21	23
Specialisation16	22
Specialisation26	19

Insight: We can observe that Specialisation4 and Specialisation7 are experiencing a relatively higher number of returns compared to other Specialisations within Department1 and Formulation as 'Form1'.

Conclusion:

Based on the patterns and trends gained from the analysis of the dataset, the following conclusions can be drawn:

1. Focus on Subcategories: The subcategories "INJECTIONS" and "TABLETS & CAPSULES" require special attention due to their higher counts of returned drug names. The hospital should conduct a thorough analysis of these subcategories to identify the underlying causes and take necessary steps to address customer satisfaction, product quality, or other issues contributing to returns.

2. Evaluation of Formulation: The "Form1" formulation stands out with the highest return counts in both the "INJECTIONS" and "TABLETS & CAPSULES" subcategories. It is essential to thoroughly evaluate this formulation, considering factors such as product effectiveness, potential side effects, and customer preferences. Improvements in the formulation or alternative options should be explored to reduce returns.

3. Supplier Assessment: Assess the performance of suppliers providing products to Department1, particularly those supplying the "Form1" formulation. Evaluate factors such as product quality, reliability, and adherence to customer requirements. Consider alternative suppliers or negotiate with existing ones to improve the quality and availability of these products.

4. Inventory Management: Ensure efficient inventory management for Department1, particularly for products with the "Form1" formulation. Optimize stock levels, expiration dates, and replenishment processes to minimize instances of expired or obsolete products. Proper inventory management can help reduce returns and maintain a more cost-effective inventory.

5. Efficient Return Management: Department1, representing the Pharmacy department, experiences a relatively higher count of return occurrences. The hospital should focus on implementing efficient return management practices, such as streamlining return processes, improving inventory management, and addressing any issues specific to Department1 that contribute to returns.

6. Specialisation-specific Strategies: Specialisation4 and Specialisation7 within Department1 and associated with the "Form1" formulation exhibit a higher number of returns. The hospital should collaborate with healthcare professionals in these specialisations to understand the reasons behind returns, gather insights, and implement specialisation-specific strategies to reduce returns and enhance customer satisfaction.

By taking these specific conclusions into consideration, the hospital can make informed business decisions and implement targeted strategies to reduce bounce rate, improve customer satisfaction, and achieve the economic success criteria.