Medical inventory optimization

Exploratory Data Analysis (SQL) by Pratiksha Saheb

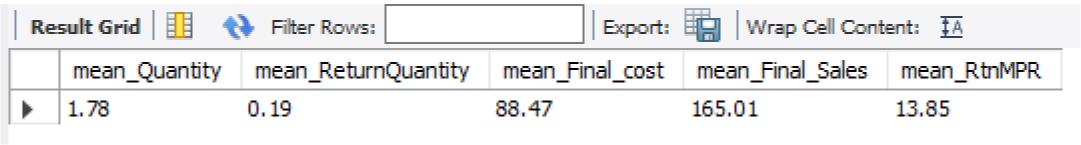
Software: MySQL Workbench

Business decisions based on the clean ‘medicines_db’ data

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

Mean :

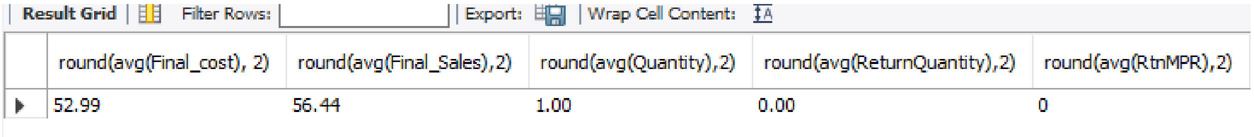
```
select
round(avg(Quantity),2) as mean_Quantity,
round(avg(ReturnQuantity),2) as mean_ReturnQuantity,
round(avg(Final_cost),2) as mean_Final_cost,
round(avg(Final_Sales),2) as mean_Final_Sales,
round(avg(RtnMPR),2) as mean_RtnMPR
from
new_table;
```



mean_Quantity	mean_ReturnQuantity	mean_Final_cost	mean_Final_Sales	mean_RtnMPR
1.78	0.19	88.47	165.01	13.85

Median:

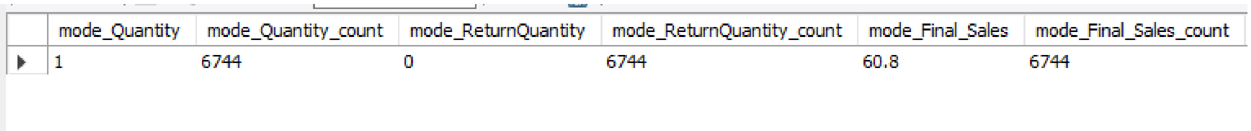
```
with ranked as
(
select Final_cost,

Final_Sales,

Quantity,

ReturnQuantity,

RtnMPR,

row_number() over (order by Final_cost) as r,

count(*) over () as c

from new_table),

median as

(

select Final_cost,

Final_Sales,

Quantity,

ReturnQuantity,

RtnMPR

from ranked

where r in (floor((c+1)/2),ceiling((c+1)/2))

)

select round(avg(Final_cost), 2),

round(avg(Final_Sales),2),

round(avg(Quantity),2),

round(avg(ReturnQuantity),2),

round(avg(RtnMPR),2)

from median;
```



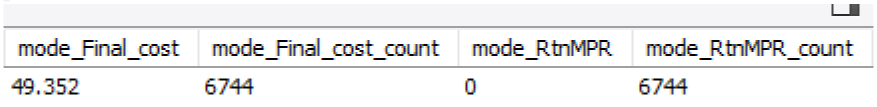
round(avg(Final_cost), 2)	round(avg(Final_Sales),2)	round(avg(Quantity),2)	round(avg(ReturnQuantity),2)	round(avg(RtnMPR),2)
52.99	56.44	1.00	0.00	0

Mode:

```
select
mode_Quantity.mode_value as mode_Quantity,
mode_Quantity.mode_count as mode_Quantity_count,
mode_ReturnQuantity.mode_value as mode_ReturnQuantity,
mode_ReturnQuantity.mode_count as mode_ReturnQuantity_count,
mode_Final_Sales.mode_value as mode_Final_Sales,
mode_Final_Sales.mode_count as mode_Final_Sales_count,
mode_Final_cost.mode_value as mode_Final_cost,
mode_Final_cost.mode_count as mode_Final_cost_count,
mode_RtnMPR.mode_value as mode_RtnMPR,
mode_RtnMPR.mode_count as mode_RtnMPR_count
from(
 SELECT Quantity AS mode_value, COUNT(*) AS mode_count
 FROM new_table
 GROUP BY Quantity
 ORDER BY COUNT(*) DESC
 LIMIT 1 ) as mode_Quantity,
 ( SELECT ReturnQuantity AS mode_value, COUNT(*) AS mode_count
 FROM new_table
 GROUP BY Quantity
 ORDER BY COUNT(*) DESC
 LIMIT 1
 ) as mode_ReturnQuantity,
 (SELECT final_cost AS mode_value, COUNT(*) AS mode_count
 FROM new_table
 GROUP BY Quantity
 ORDER BY COUNT(*) DESC
 LIMIT 1
 ) as mode_Final_cost,

 (SELECT Final_Sales AS mode_value, COUNT(*) AS mode_count

 FROM new_table

 GROUP BY Quantity

 ORDER BY COUNT(*) DESC

 LIMIT 1

 ) as mode_Final_Sales,

 ( select RtnMPR AS mode_value, COUNT(*) AS mode_count

 FROM new_table

 GROUP BY Quantity

 ORDER BY COUNT(*) DESC

 LIMIT 1

 ) as mode_RtnMPR;
```



mode_Quantity	mode_Quantity_count	mode_ReturnQuantity	mode_ReturnQuantity_count	mode_Final_Sales	mode_Final_Sales_count
1	6744	0	6744	60.8	6744

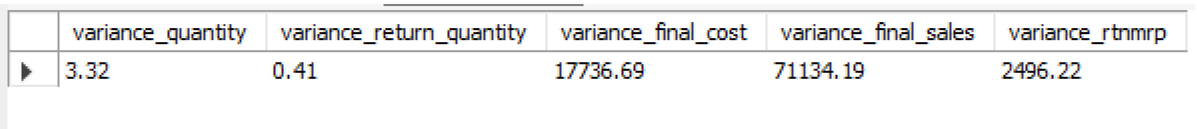


mode_Final_cost	mode_Final_cost_count	mode_RtnMPR	mode_RtnMPR_count
49.352	6744	0	6744

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

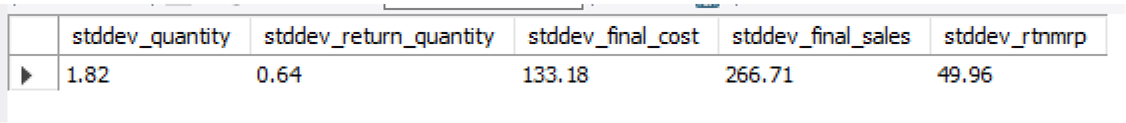
Variance:

```
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(RtnMPR), 2) AS variance_rtnmrp

FROM new_table;
```



	variance_quantity	variance_return_quantity	variance_final_cost	variance_final_sales	variance_rtnmrp
▶	3.32	0.41	17736.69	71134.19	2496.22

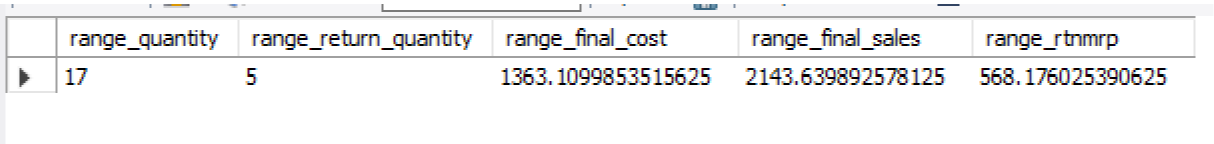
```
SELECT
 ROUND(STDDEV(Quantity), 2) AS stddev_quantity,
 ROUND(STDDEV(ReturnQuantity), 2) AS stddev_return_quantity,
 ROUND(STDDEV(Final_Cost), 2) AS stddev_final_cost,
 ROUND(STDDEV(Final_Sales), 2) AS stddev_final_sales,
 ROUND(STDDEV(RtnMPR), 2) AS stddev_rtnmrp
FROM new_table;
```



	stddev_quantity	stddev_return_quantity	stddev_final_cost	stddev_final_sales	stddev_rtnmrp
▶	1.82	0.64	133.18	266.71	49.96

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(RtnMPR) - MIN(RtnMPR) AS range_rtnmrp
FROM new_table;
```

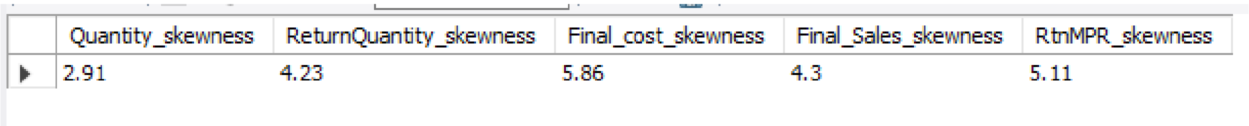


	range_quantity	range_return_quantity	range_final_cost	range_final_sales	range_rtnmrp
▶	17	5	1363.1099853515625	2143.639892578125	568.176025390625

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

Skewness:

```
select Quantity_skewness1.Quantity_skewness,
ReturnQuantity_skewness1.ReturnQuantity_skewness,
Final_cost_skewness1.Final_cost_skewness,
Final_Sales_skewness1.Final_Sales_skewness,
RtnMPR_skewness1.RtnMPR_skewness
from
(SELECT
ROUND((SUM(POW(Quantity - (SELECT AVG(Quantity) FROM new_table), 3)) / (COUNT(*) *
POW(STDDEV(Quantity), 3))), 2) AS Quantity_skewness
FROM new_table) Quantity_skewness1,
(select
round((sum(pow(ReturnQuantity - (select avg(ReturnQuantity) from new_table),3))/(count(*) *
pow(stddev(ReturnQuantity),3))),2) as  ReturnQuantity_skewness from new_table)
ReturnQuantity_skewness1,
(select
round((sum(pow(Final_cost - (select avg(Final_cost) from new_table),3))/(count(*) *
pow(stddev(Final_cost),3))),2) as  Final_cost_skewness from new_table)
Final_cost_skewness1,
 (
select
round((sum(pow(Final_Sales - (select avg(Final_Sales) from new_table),3))/(count(*) *
pow(stddev(Final_Sales),3))),2) as  Final_Sales_skewness from new_table)
Final_Sales_skewness1,
(select
round((sum(pow(RtnMPR - (select avg(RtnMPR) from new_table),3))/(count(*) *
pow(stddev(RtnMPR),3))),2) as  RtnMPR_skewness from new_table)
RtnMPR_skewness1;
```

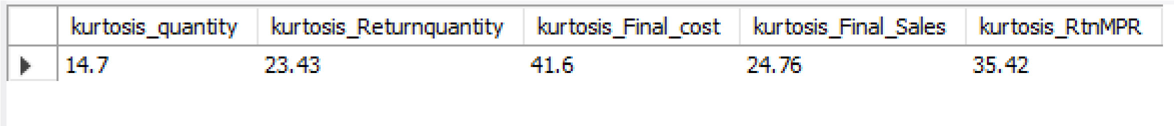


	Quantity_skewness	ReturnQuantity_skewness	Final_cost_skewness	Final_Sales_skewness	RtnMPR_skewness
▶	2.91	4.23	5.86	4.3	5.11

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

Kurtosis:

```
select
kurtosis_quantity1.kurtosis_quantity,
kurtosis_Returnquantity1.kurtosis_Returnquantity,
kurtosis_Final_cost1.kurtosis_Final_cost,
kurtosis_Final_Sales1.kurtosis_Final_Sales,
kurtosis_RtnMPR1.kurtosis_RtnMPR
from
(SELECT
ROUND((SUM(POWER(Quantity - (SELECT AVG(Quantity) FROM new_table), 4)) / (COUNT(Quantity) * POWER(STDDEV(Quantity),
4))), 2) AS kurtosis_quantity
from new_table) kurtosis_quantity1,
(SELECT
ROUND((SUM(POWER(ReturnQuantity - (SELECT AVG(ReturnQuantity) FROM new_table), 4)) / (COUNT(ReturnQuantity) * POWER(STDDEV(ReturnQuantity),
4))), 2) AS kurtosis_Returnquantity
from new_table) kurtosis_Returnquantity1,
(SELECT
ROUND((SUM(POWER(Final_cost - (SELECT AVG(Final_cost) FROM new_table), 4)) / (COUNT(Final_cost) * POWER(STDDEV(Final_cost),
4))), 2) AS kurtosis_Final_cost
from new_table) kurtosis_Final_cost1,
(SELECT
ROUND((SUM(POWER(Final_Sales - (SELECT AVG(Final_Sales) FROM new_table), 4)) / (COUNT(Final_Sales) * POWER(STDDEV(Final_Sales),
4))), 2) AS kurtosis_Final_Sales
from new_table) kurtosis_Final_Sales1,
(SELECT
ROUND((SUM(POWER(RtnMPR - (SELECT AVG(RtnMPR) FROM new_table), 4)) / (COUNT(RtnMPR) *
POWER(STDDEV(RtnMPR),

4))), 2) AS kurtosis_RtnMPR

from new_table) kurtosis_RtnMPR1;
```



	kurtosis_quantity	kurtosis_Returnquantity	kurtosis_Final_cost	kurtosis_Final_Sales	kurtosis_RtnMPR
▶	14.7	23.43	41.6	24.76	35.42

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

Parameter	**Uncleaned Data**	**Cleaned_Data**
Mean	**Quantity -2.25**	**Quantity-1.78**
	ReturnQuantity -0.30	**ReturnQuantity -0.19**
	Final_cost-124.43	**Final_cost-88.47**
	Final_Sales-232.79	**Final_Sales-165.01**
	RtnMPR-29.95	**RtnMPR-13.85**
Median	**Quantity-3.00**	**Quantity-1.00**
	ReturnQuantity -0.00	**ReturnQuantity -0.00**
	Final_cost-141	**Final_cost-52.99**
	Final_Sales-53.57	**Final_Sales-56.44**
	RtnMPR-0-00	**RtnMPR-0.00**
Mode	**Quantity-1**	**Quantity-1**
	ReturnQuantity -0	**ReturnQuantity -0**
	Final_cost-49.352	**Final_cost-49.352**
	Final_Sales-60.8	**Final_Sales-60.8**
	RtnMPR-0	**RtnMPR-0**
Variance	**Quantity-27.14**	**Quantity-3.32**
	ReturnQuantity 2.17	**ReturnQuantity -0.41**
	Final_cost-221685.41	**Final_cost-17736.69**
	Final_Sales-458223.23	**Final_Sales-71134.19**
	RtnMPR-34178.39	**RtnMPR-2596.22**
StandardDeviation	**Quantity-5.21**	**Quantity-1.82**
	ReturnQuantity -1.67	**ReturnQuantity -0.64**
	Final_cost-470.83	**Final_cost-133.18**
	Final_Sales-676.93	**Final_Sales-266.71**
	RtnMPR-184.87	**RtnMPR-49.96**
Range	**Quantity-150**	**Quantity-17**
	ReturnQuantity -50	**ReturnQuantity -5**
	Final_cost-33138	**Final_cost-1363.11**
	Final_Sales-39490	**Final_Sales-2143.64**

	RtnMPR-8014	**RtnMPR-568.18**
Skewness	**Quantity-11.27**	**Quantity-2.91**
	ReturnQuantity -16.88	**ReturnQuantity -4.23**
	Final_cost-34.22	**Final_cost-5.86**
	Final_Sales-21.09	**Final_Sales-4.3**
	RtnMPR-15.58	**RtnMPR5.11**
Kurtosis	**Quantity-180.5**	**Quantity-14.7**
	ReturnQuantity -398.03	**ReturnQuantity -23.43**
	Final_cost-1984.44	**Final_cost-41.6**
	Final_Sales-947.71	**Final_Sales-24.76**
	RtnMPR-395.28	**RtnMPR-35.42**

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.