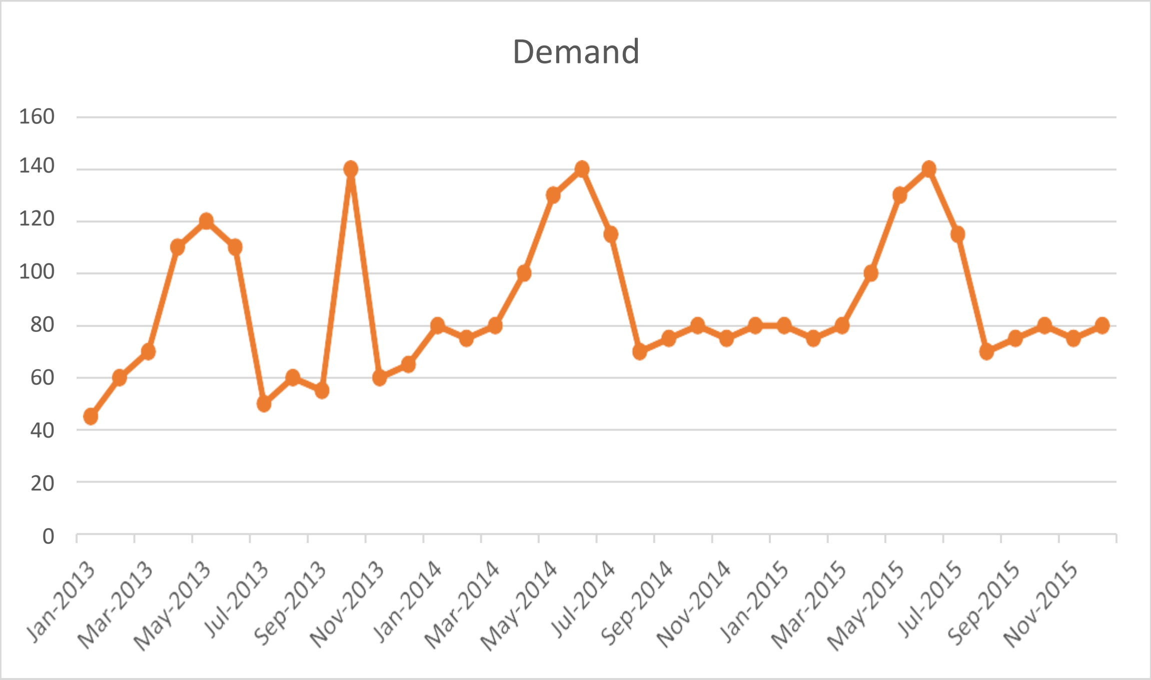
Question 1 Data Analysis and Interpretation

What can be said or inferred from the below Time series? (Please limit it to 5 inferences)

Refer to Excel for Data

- Can use Excel/any other data analysis tools
- List out any assumptions made which are not given in the question
- Can use graphs as a means to convey the results (only if required)



Question 2 SQL and Use

INPUT (TABLE1)		
VELOCITY	**VOLUME**	**SALES**
A	VERY HIGH	613
B	VERY HIGH	672
C	VERY HIGH	834
D	VERY HIGH	416
E	VERY HIGH	96
A	HIGH	170
B	HIGH	827
C	HIGH	778
D	HIGH	692
E	HIGH	562
A	MEDIUM-HIGH	731
B	MEDIUM-HIGH	127
C	MEDIUM-HIGH	848
D	MEDIUM-HIGH	335
E	MEDIUM-HIGH	465
A	MEDIUM-LOW	399
B	MEDIUM-LOW	877
C	MEDIUM-LOW	867
D	MEDIUM-LOW	987
E	MEDIUM-LOW	784
A	LOW	848
B	LOW	626
C	LOW	28
D	LOW	246
E	LOW	166
A	VERY LOW	626
B	VERY LOW	150
C	VERY LOW	821
D	VERY LOW	198
E	VERY LOW	622

Directions

1. Take the input table (TABLE1) and write ANSI-Standard SQL to produce the output below (TABLE2). Database-specific functions must be avoided.
2. TABLE1 field data types are as follows:
   1) VELOCITY VARCHAR(1): The rate at which a product sells.
   2) VOLUME VARCHAR(20): A classification for a store to determine how much product is sold relative to its peers.
   3) SALES INTEGER: Count of units sold.
3. May use sub-queries if needed but everything must be self-contained in a single executable query.
4. Avoid using Database specific functions (like Pivot in SQL server is Database specific)
5. Syntactical Example:

```
SELECT SUM(TBL1.SALES)
FROM TABLE1   TBL1
WHERE VELOCITY = ‘A’
```

RETURNS **3387**

OUTPUT (TABLE2)							
VELOCITY	**VERY HIGH**	**HIGH**	**MEDIUM-HIGH**	**MEDIUM-LOW**	**LOW**	**VERY LOW**	**TOTAL**
A	613	170	731	399	848	626	3387
B	672	827	127	877	626	150	3279
C	834	778	848	867	28	821	4176
D	416	692	335	987	246	198	2874
E	96	562	465	784	166	622	2695
TOTAL	2631	3029	2506	3914	1914	2417	16411

Question 3 SQL and Use: Critical Thinking (Geospatial)

INPUT (TABLE1)

LOCATION_ID	LATITUDE	LONGITUDE
1	40.9283	-98.3821
2	38.953	-94.5342
3	33.925	-116.956

OUTPUT (TABLE2)

ORIG_LOC_ID	DEST_LOC_ID	LOS_DSTNC
1	1	0
1	2	245
1	3	1124
2	1	245
2	2	0
2	3	1289
3	1	1124
3	2	1289
3	3	0

Directions

1. Take the input table (TABLE1) and write ANSI-Standard SQL to produce the output below (TABLE2). LOS_DSTCN is the line-of-sight or "as-the-crow-flies" distance (in miles) between the Origin location (ORIG_LOC_ID) and the Destination location (DEST_LOC_ID). Database-specific functions must be avoided.
2. TABLE1 field data types are as follows:
    1) LOCATION_ID INTEGER: A field that provides a unique identifier to a physical location.
    2) LATITUDE DECIMAL (11,7): The latitudinal coordinate for the physical location.
    3) LONGITUDE DECIMAL (11,7): The longitudinal coordinate for the physical location.
3. May use sub-queries if needed but everything must be self-contained in a single executable query.
4. Avoid using Database specific functions (like Pivot in SQL server is Database specific)
5. Syntactical Example:
   SELECT TBL1.LATITUDE
   FROM TABLE1 TBL1
   WHERE LOCATION_ID = 1

   Returns **40.9283**

Question 4 SQL and Use

INPUT (TABLE1)		
PRODUCT	**WEEKNUMBER**	**SALES**
HAMMER	1	17
HAMMER	2	20
HAMMER	3	17
HAMMER	4	10
HAMMER	5	12
HAMMER	6	13
HAMMER	7	2
HAMMER	8	25
SINK	1	25
SINK	2	20
SINK	3	9
SINK	4	7
SINK	5	24
SINK	6	16

Directions

1. Take the input table (TABLE1) and write ANSI-Standard SQL to produce the output below (TABLE2). BEST_CONSEC_4WEEKS is a character string that tells the actual week numbers with the highest sum of sales over a consecutive 4 week period. BEST_CONSEC_4WEEKS_SLS provides the total units sold for that product during the weeks identified above. Database-specific functions must be avoided.
2. TABLE1 field data types are as follows:
    1) PRODUCT VARCHAR(20): The product sold in the store.
    2) WEEKNUMBER INTEGER: Fiscal week number.
    3) SALES INTEGER: Count of units sold on that week.
3. May use sub-queries if needed but everything must be self-contained in a single executable query.
4. Avoid using Database specific functions (like Pivot in SQL server is Database specific)
5. Syntactical Example:

    ```
    SELECT SUM(TBL1.SALES)
    FROM TABLE1  TBL1
    WHERE WEEKNUMBER = 1
    ```

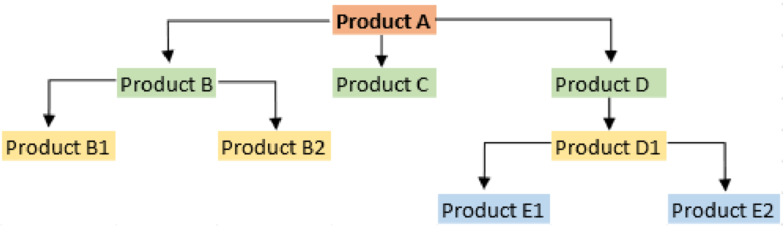
    Returns **64**

SINK	7	10
SINK	8	16
BUCKET	1	22
BUCKET	2	2
BUCKET	3	10
BUCKET	4	24
BUCKET	5	9
BUCKET	6	20
BUCKET	7	9
BUCKET	8	21

OUTPUT (TABLE2)		
PRODUCT	**BEST_CONSEC_4WEEKS**	**BEST_CONSEC_4WEEKS_SLS**
HAMMER	1-4	64
SINK	5-8	66
BUCKET	3-6	63

Question 5 SQL and Use

Following is the Bill of Materials for Product A



The same is stored in a Table (Just 2 fields– ChildProduct and ParentProduct) in a SQL Database in the following manner:

Input Product Table

ChildProduct	ParentProduct
Product B	Product A
Product C	Product A
Product D	Product A
Product B1	Product B
Product B2	Product B
Product D1	Product D
Product E1	Product D1
Product E2	Product D1

Using SQL, generate the following output from the above table.

- Do not use database specific functions/hardcode products.
- Syntax is not important, but the approach is

Output

ChildProduct	ParentProduct
Product B	Product A
Product B1	Product A
Product B2	Product A
Product C	Product A
Product D	Product A
Product D1	Product A
Product E1	Product A
Product E2	Product A
Product B1	Product B
Product B2	Product B
Product D1	Product D
Product E1	Product D
Product E2	Product D

Product E1	Product D1
Product E2	Product D1

Each of the Parent Product should have all the Child product (and Child of child) tied to them.

Question 6 SQL and Use

SHIPMENT TABLE

Shipment ID	Ship Date	Origin	Destination
10001	2/3/2015	RDC	Store 1
10002	2/5/2015	RDC	Store 2
10004	2/6/2015	RDC	Store 1
10007	2/6/2015	RDC	Store 2
10008	2/6/2015	RDC	Store 3
10009	2/7/2015	RDC	Store 3
10011	2/8/2015	RDC	Store 2
10013	2/9/2015	RDC	Store 1
10014	2/11/2015	RDC	Store 2

RECEIPT TABLE

Receipt Date	Origin	Destination
2/3/2015	RDC	Store 1
2/6/2015	RDC	Store 2
2/8/2015	RDC	Store 1
2/7/2015	RDC	Store 2
2/8/2015	RDC	Store 3
2/8/2015	RDC	Store 3
2/10/2015	RDC	Store 2
2/10/2015	RDC	Store 1
2/12/2015	RDC	Store 2

The above two input tables have the Shipment Data and Receipt Data respectively. The objective is to tie these two data together to appear like the below Output table. Each Shipment has a corresponding Receipt, but not necessarily on the same day. For example, Shipment 10002 that was shipped on 2/5/2015 from RDC to Store 2 has a Receipt Date of 2/6/2015. The total number of shipments for a given RDC – Store combination is equal to the total number of receipts for the same combination.

Hint – Assume the shipment were received in the order they were shipped. In cases where there were 2 Receipts for the same store on the same day, randomly assign the shipments to the 2 receipts.

OUTPUT TABLE

Shipment ID	Ship Date	Receipt Date	Origin	Destination
10001	2/3/2015	2/3/2015	RDC	Store 1
10002	2/5/2015	2/6/2015	RDC	Store 2
10004	2/6/2015	2/8/2015	RDC	Store 1
10007	2/6/2015	2/7/2015	RDC	Store 2
10008	2/6/2015	2/8/2015	RDC	Store 3
10009	2/7/2015	2/8/2015	RDC	Store 3
10011	2/8/2015	2/10/2015	RDC	Store 2
10013	2/9/2015	2/10/2015	RDC	Store 1
10014	2/11/2015	2/12/2015	RDC	Store 2

Syntax is not important, but the approach is.

Can use database specific only if necessary.