Table/Dataframe (depending on language) *Sales\_data* has the following sample rows (input csv shared separately)

|  |  |  |
| --- | --- | --- |
| **Sales\_data** | | |
| Month | Store | Sales |
| 01/01/19 | A | 100 |
| 01/02/19 | A | 300 |
| 01/03/19 | A | 500 |
| 01/01/19 | B | 50 |
| 01/02/19 | B | 300 |
| 01/03/19 | B | 50 |
| 01/01/19 | C | 200 |
| 01/02/19 | C | 200 |
| 01/03/19 | C | 100 |
| 01/01/19 | D | 150 |
| 01/02/19 | D | 100 |
| 01/03/19 | D | 200 |

## Question 1)

Write a Code to extract the maximum sales in each month along with the store which had maximum sales. In case of a tie in a given, retain all stores with maximum sales

PFB sample output:

|  |  |  |
| --- | --- | --- |
| Max\_sales\_data | | |
| Month | Max\_Sales | Store\_max |
| 01/01/19 | 200 | C |
| 01/02/19 | 300 | B |
| 01/02/19 | 300 | A |
| 01/03/19 | 500 | A |

Attempt in both SQL/Python

## SQL Solution Question 1)

## Python Solution Question 1)

Question2) Using the same dataset as earlier

Print the Numbers of stores with Sales greater than 100 in each month

Expected output

|  |  |
| --- | --- |
| Month | Sales\_gt\_100 |
| 01/01/19 | 2 |
| 01/02/19 | 3 |
| 01/03/19 | 2 |

Attempt in both SQL/Python

## SQL Solution Question 2)

## Python Solution Question 2)

Question3) In the Overall Data bucket sales into three buckets(irrespective of month) and summarize

Expected output

|  |  |
| --- | --- |
| Sales\_bucket | Count\_Stores |
| A.0-150 | 6 |
| B.151-299 | 3 |
| C.ge 300 | 3 |

Attempt in both SQL/Python

## SQL Solution Question 3)

## Python Solution Question 3)

Question4) For each store comment whether sales are monotonically increasing month on month for each store

Monotonically increasing is defined as

Sales(01/03/19)> Sales(01/02/19)>Sales(01/01/19)

Expected output

|  |  |
| --- | --- |
| **Store** | **Mono\_flag** |
| A | 1 |
| B | 0 |
| C | 0 |
| D | 0 |

Attempt in Python

## Python Solution Question 4)