Indian Statistical Institute, Kolkata

Assignment-I

Student :
Snehashish Ghosh, CS2431

Teacher :
Prof. Malay Bhattacharyya

January 15, 2025

Contents

1 Question 1

In [1]:
```
import pandas as pd
import numpy as np
import random
import sklearn.datasets as datasets
import sys

sys.path.insert(1, '..')
from check_data_consistency import DataConsistencyChecker
```

In [2]:
```
pd.options.display.max_columns = 1000
pd.options.display.max_colwidth = 1000
pd.options.display.max_rows = 1000
pd.options.display.width = 10000
```

In [11]:
```
data = datasets.fetch_california_housing()
df = pd.read_csv("power consumption.csv") #https://www.kaggle.com/datasets/sauga
df.head(2)
```

Out[11]:

	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1	Zone
0	01-01-2017 00:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.8753
1	01-01-2017 00:10	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.0759

In [12]:
```
dc = DataConsistencyChecker()
dc.init_data(df)
```

In [13]:
```
# Run a small set of tests to start. In this example, we run a single test.

_ = dc.check_data_quality()
```

```
Executing test   0: MISSING_VALUES
Executing test   1: RARE_VALUES
Executing test   2: UNIQUE_VALUES
Executing test   3: PREV_VALUES_DT
Executing test   4: MATCHED_MISSING
Executing test   5: OPPOSITE_MISSING
Executing test   6: SAME_VALUES
Executing test   7: SAME_OR_CONSTANT
Executing test   8: UNIQUE_PAIR
Executing test   9: POSITIVE
Executing test  10: NEGATIVE
Executing test  11: NUMBER_DECIMALS
Executing test  12: RARE_DECIMALS
Executing test  13: COLUMN_ORDERED_ASC
Executing test  14: COLUMN_ORDERED_DESC
Executing test  15: COLUMN_TENDS_ASC
Executing test  16: COLUMN_TENDS_DESC
Executing test  17: SIMILAR_PREVIOUS
Executing test  18: UNUSUAL_ORDER_MAGNITUDE
Executing test  19: FEW_NEIGHBORS
Executing test  20: FEW_WITHIN_RANGE
Executing test  21: VERY_SMALL
Executing test  22: VERY_LARGE
Executing test  23: VERY_SMALL_ABS
Executing test  24: MULTIPLE_OF_CONSTANT
Executing test  25: ROUNDING
Executing test  26: NON_ZERO
Executing test  27: LESS_THAN_ONE
Executing test  28: GREATER_THAN_ONE
Executing test  29: INVALID_NUMBERS
Executing test  30: LARGER_DIFF_RANGE
Executing test  31: LARGER_SAME_RANGE
Executing test  32: MUCH_LARGER
Executing test  33: SIMILAR_WRT_RATIO
Executing test  34: SIMILAR_WRT_DIFF
Executing test  35: SIMILAR_TO_INVERSE
Executing test  36: SIMILAR_TO_NEGATIVE
Executing test  37: CONSTANT_SUM
Executing test  38: CONSTANT_DIFF
Executing test  39: CONSTANT_PRODUCT
Executing test  40: CONSTANT_RATIO
Executing test  41: EVEN_MULTIPLE
Executing test  42: RARE_COMBINATION
Executing test  43: CORRELATED_NUMERIC
Executing test  44: MATCHED_ZERO
Executing test  45: OPPOSITE_ZERO
Executing test  46: RUNNING_SUM
Executing test  47: A_ROUNDED_B
Executing test  48: MATCHED_ZERO_MISSING
Executing test  49: SIMILAR_TO_DIFF
Executing test  50: SIMILAR_TO_PRODUCT
Executing test  51: SIMILAR_TO_RATIO
Executing test  52: LARGER_THAN_SUM
Executing test  53: SUM_OF_COLUMNS
Executing test  54: MEAN_OF_COLUMNS
Executing test  55: MIN_OF_COLUMNS
Executing test  56: MAX_OF_COLUMNS
Executing test  57: MATCHED_SET_POS_NEG
Executing test  58: MATCHED_SET_ZERO_NON_ZERO
Executing test  59: DECISION_TREE_REGRESSOR
```

```
Executing test  60: LINEAR_REGRESSION
Executing test  61: PREDICT_NULL_DT
Executing test  62: EARLY_DATES
Executing test  63: LATE_DATES
Executing test  64: UNUSUAL_DAY_OF_WEEK
Executing test  65: UNUSUAL_DAY_OF_MONTH
Executing test  66: UNUSUAL_MONTH
Executing test  67: UNUSUAL_HOUR
Executing test  68: UNUSUAL_MINUTES
Executing test  69: CONSTANT_DOM
Executing test  70: CONSTANT_LAST_DOM
Executing test  71: CONSTANT_GAP
Executing test  72: LARGE_GAP
Executing test  73: SMALL_GAP
Executing test  74: LATER
Executing test  75: SAME_DATE
Executing test  76: SAME_MONTH
Executing test  77: CORRELATED_DATES
Executing test  78: LARGE_GIVEN_DATE
Executing test  79: SMALL_GIVEN_DATE
Executing test  80: BINARY_SAME
Executing test  81: BINARY_OPPOSITE
Executing test  82: BINARY_IMPLIES
Executing test  83: BINARY_AND
Executing test  84: BINARY_OR
Executing test  85: BINARY_XOR
Executing test  86: BINARY_NUM_SAME
Executing test  87: BINARY_RARE_COMBINATION
Executing test  88: BINARY_MATCHES_VALUES
Executing test  89: BINARY_TWO_OTHERS_MATCH
Executing test  90: BINARY_MATCHES_SUM
Executing test  91: BLANK_VALUES
Executing test  92: LEADING_WHITESPACE
Executing test  93: TRAILING_WHITESPACE
Executing test  94: FIRST_CHAR_ALPHA
Executing test  95: FIRST_CHAR_NUMERIC
Executing test  96: FIRST_CHAR_SMALL_SET
Executing test  97: FIRST_CHAR_UPPERCASE
Executing test  98: FIRST_CHAR_LOWERCASE
Executing test  99: LAST_CHAR_SMALL_SET
Executing test 100: COMMON_SPECIAL_CHARS
Executing test 101: COMMON_CHARS
Executing test 102: NUMBER_ALPHA_CHARS
Executing test 103: NUMBER_NUMERIC_CHARS
Executing test 104: NUMBER_ALPHANUMERIC_CHARS
Executing test 105: NUMBER_NON-ALPHANUMERIC_CHARS
Executing test 106: NUMBER_CHARS
Executing test 107: NONPRINTABLE_CHARS
Executing test 108: MANY_CHARS
Executing test 109: FEW_CHARS
Executing test 110: POSITION_NON-ALPHANUMERIC
Executing test 111: CHARS_PATTERN
Executing test 112: UPPERCASE
Executing test 113: LOWERCASE
Executing test 114: CHARACTERS_USED
Executing test 115: FIRST_WORD_SMALL_SET
Executing test 116: LAST_WORD_SMALL_SET
Executing test 117: NUMBER_WORDS
Executing test 118: LONGEST_WORDS
Executing test 119: COMMON_WORDS
```

```
Executing test 120: RARE_WORDS
Executing test 121: GROUPED_STRINGS
Executing test 122: RARE_PAIRS
Executing test 123: RARE_PAIRS_FIRST_CHAR
Executing test 124: RARE_PAIRS_FIRST_WORD
Executing test 125: RARE_PAIRS_FIRST_WORD_VAL
Executing test 126: SIMILAR_CHARACTERS
Executing test 127: SIMILAR_NUM_CHARS
Executing test 128: SIMILAR_WORDS
Executing test 129: SIMILAR_NUM_WORDS
Executing test 130: SAME_FIRST_CHARS
Executing test 131: SAME_FIRST_WORD
Executing test 132: SAME_LAST_WORD
Executing test 133: SAME_ALPHA_CHARS
Executing test 134: SAME_NUMERIC_CHARS
Executing test 135: SAME_SPECIAL_CHARS
Executing test 136: A_PREFIX_OF_B
Executing test 137: A_SUFFIX_OF_B
Executing test 138: B_CONTAINS_A
Executing test 139: CORRELATED_ALPHA_ORDER
Executing test 140: LARGE_GIVEN_VALUE
Executing test 141: SMALL_GIVEN_VALUE
Executing test 142: LARGE_GIVEN_PREFIX
Executing test 143: SMALL_GIVEN_PREFIX
Executing test 144: GROUPED_STRINGS_BY_NUMERIC
Executing test 145: LARGE_GIVEN_PAIR
Executing test 146: SMALL_GIVEN_PAIR
Executing test 147: CORRELATED_GIVEN_VALUE
Executing test 148: DECISION_TREE_CLASSIFIER
Executing test 149: C_IS_A_OR_B
Executing test 150: TWO_PAIRS
Executing test 151: UNIQUE_SETS_VALUES
Executing test 152: MISSING_VALUES_PER_ROW
Executing test 153: ZERO_VALUES_PER_ROW
Executing test 154: UNIQUE_VALUES_PER_ROW
Executing test 155: NEGATIVE_VALUES_PER_ROW
Executing test 156: SMALL_AVG_RANK_PER_ROW
Executing test 157: LARGE_AVG_RANK_PER_ROW

Data consistency check complete.
Analysed 52,416 rows, 9 columns
Executed 158 tests.

Patterns without Exceptions:
Found 45 patterns without exceptions
11 tests (6.96% of tests) identified at least one pattern without exceptions eac
h.
By default some patterns are not listed in calls to display_detailed_results().

Patterns with Exceptions:
Found 9 patterns with exceptions
4 tests (2.53% of tests) flagged at least one exception each.
Flagged 611 row(s) with at least one exception.
Flagged 7 column(s) with at least one exception.
```

In [14]:
```
# In the next few cells, we look at the output of the tests.

dc.summarize_patterns_and_exceptions()
```

Out[14]:

	Test ID	Number Patterns without Exceptions	Number Patterns with Exceptions
0	MISSING_VALUES	9	
1	UNIQUE_VALUES	1	
2	UNIQUE_PAIR		1
3	POSITIVE	8	
4	NUMBER_DECIMALS	7	
5	COLUMN_ORDERED_ASC	1	
6	COLUMN_TENDS_ASC	1	
7	SIMILAR_PREVIOUS	3	5
8	FEW_WITHIN_RANGE		1
9	NON_ZERO	8	
10	GREATER_THAN_ONE	5	
11	LARGER_DIFF_RANGE		2
12	MISSING_VALUES_PER_ROW	1	
13	NEGATIVE_VALUES_PER_ROW	1	

In [15]:
```
# Run a small set of tests to start. In this example, we run a single test.

_ = dc.check_data_quality(execute_list=['SIMILAR_PREVIOUS'])
```

```
Executing test  17: SIMILAR_PREVIOUS

Data consistency check complete.
Analysed 52,416 rows, 9 columns
Executed 1 tests.

Patterns without Exceptions:
Found 3 patterns without exceptions
1 tests (100.00% of tests) identified at least one pattern without exceptions eac
h.
By default some patterns are not listed in calls to display_detailed_results().

Patterns with Exceptions:
Found 5 patterns with exceptions
1 tests (100.00% of tests) flagged at least one exception each.
Flagged 165 row(s) with at least one exception.
Flagged 5 column(s) with at least one exception.
```

In [16]:
```
dc.quick_report()
```

Patterns List (short list only)

	Test ID	Column(s)	Description of Pattern	Pattern ID
0	SIMILAR_PREVIOUS	Temperature	The values in "Temperature" are consistently similar to the previous value, more so than they are si...	0
1	SIMILAR_PREVIOUS	Zone 3	The values in "Zone 3 " are consistently similar to the previous value, more so than they are simil...	1
2	SIMILAR_PREVIOUS	DateTime	The values in "DateTime" are consistently similar to the previous value, more so than they are simil...	2

Patterns by Test and Feature

Test ID	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1	Z
SIMILAR_PREVIOUS	✔	✔						

Exceptions List

	Test ID	Column(s)	Description of Pattern	Number of Exceptions	Issue ID
0	SIMILAR_PREVIOUS	Humidity	The values in "Humidity" are consistently similar to the previous value, more so than they are simil...	2	0
1	SIMILAR_PREVIOUS	Wind Speed	The values in "Wind Speed" are consistently similar to the previous value, more so than they are sim...	1	1
2	SIMILAR_PREVIOUS	general diffuse flows	The values in "general diffuse flows" are consistently similar to the previous value, more so than t...	160	2
3	SIMILAR_PREVIOUS	Zone 1	The values in "Zone 1" are consistently similar to the previous value, more so than they are similar...	1	3
4	SIMILAR_PREVIOUS	Zone 2	The values in "Zone 2 " are consistently similar to the previous value, more so than they are simil...	1	4

Exceptions Summary by Test and Feature

Test ID	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1	Z
SIMILAR_PREVIOUS			2	1	160		1	

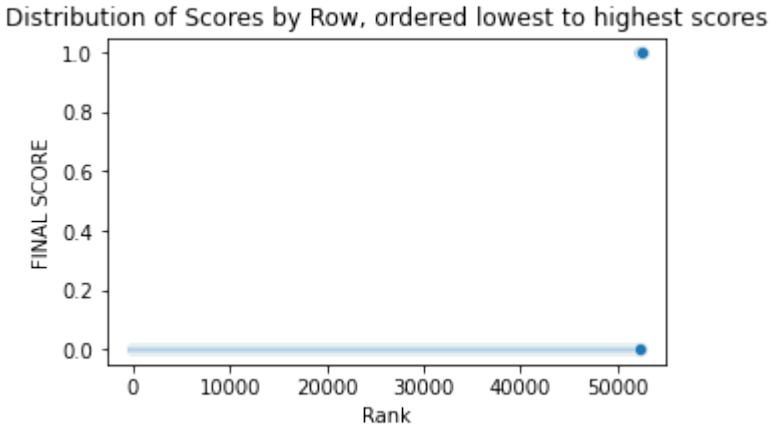
Exceptions Summary by Test

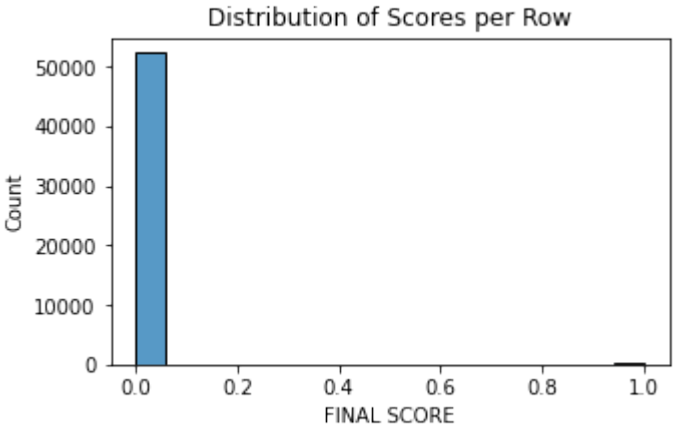
Test ID	Number of Columns Flagged At Least Once	Number of Issues Total
SIMILAR_PREVIOUS	5	165

Summary of Patterns and Exceptions (all tests)

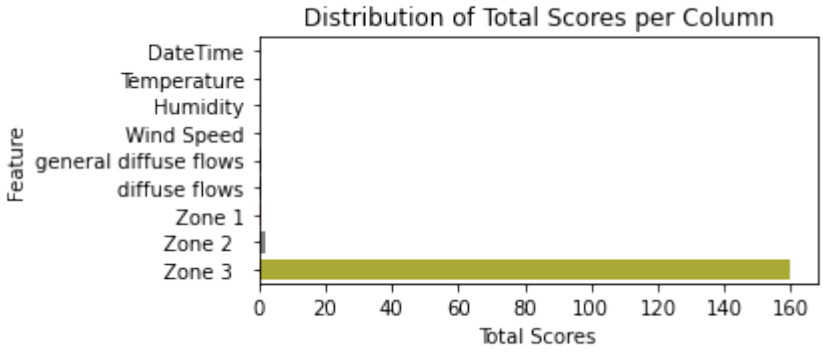
	Test ID	Number Patterns without Exceptions	Number Patterns with Exceptions
0	SIMILAR_PREVIOUS	3	5

Final Scores by Row of the Data

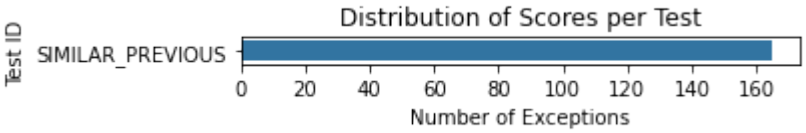




Final Scores by Feature



Final Scores by Test



In [19]:
```
dc.display_detailed_results(test_id_list=['POSITIVE'])
```

In [22]:
```
for i in range(10):
  dc.display_detailed_results(issue_id_list=[i])
```

Columns(s): Humidity

Issue ID: 0

A strong pattern, and exceptions to the pattern, were found.

Description: The values in "Humidity" are consistently similar to the previous value, more so tha similar to the median value of the column (69.86), with exceptions.

Number of exceptions: 2 (0.0038% of rows)

Examples of values NOT flagged (showing a consecutive set of rows):

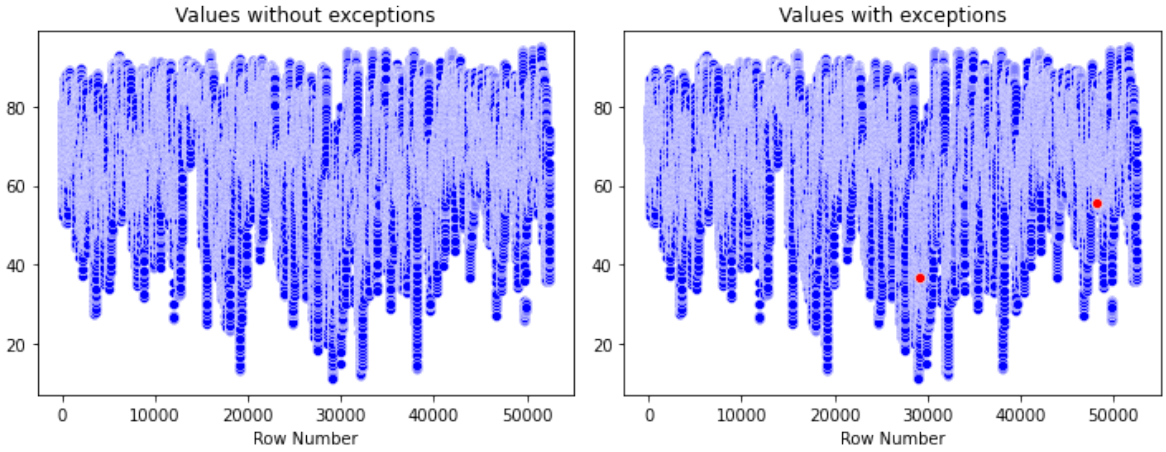
	Humidity
30403	67.660000
30404	67.860000
30405	67.800000
30406	67.200000
30407	66.930000
30408	66.970000
30409	66.730000
30410	66.530000
30411	66.370000
30412	65.400000

Flagged values:

	Humidity
29219	36.810000
48096	55.770000

Showing a flagged example (row 29219) with 5 rows before and 5 rows after (if available) the fla

	Humidity
29214	73.400000
29215	74.300000
29216	75.200000
29217	75.300000
29218	64.620000
29219	36.810000
29220	30.140000
29221	29.610000
29222	29.710000
29223	31.110000



Columns(s): Wind Speed

Issue ID: 1

A strong pattern, and exceptions to the pattern, were found.

Description: The values in "Wind Speed" are consistently similar to the previous value, more so similar to the median value of the column (0.086), with exceptions.

Number of exceptions: 1 (0.0019% of rows)

Examples of values NOT flagged (showing a consecutive set of rows):

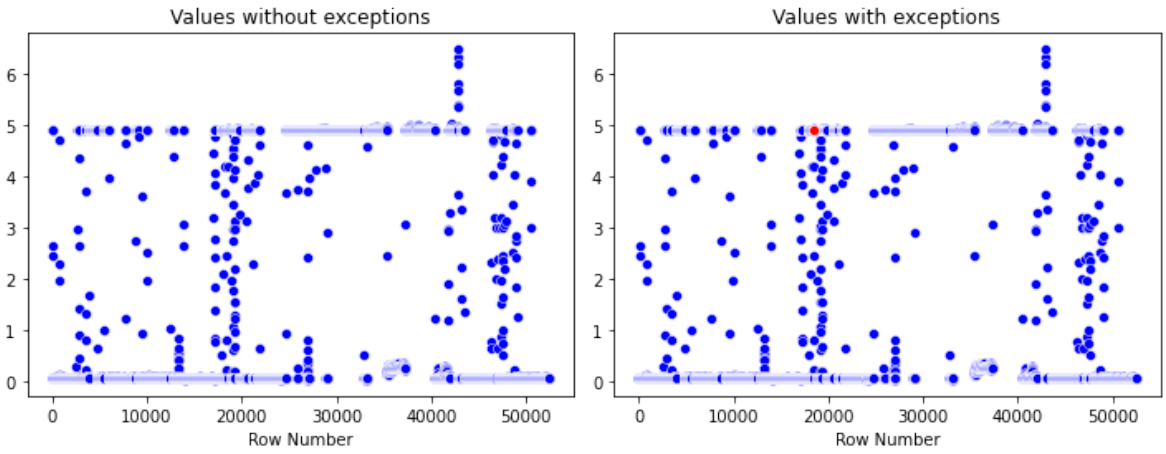
	Wind Speed
32103	4.907000
32104	4.909000
32105	4.908000
32106	4.903000
32107	4.907000
32108	4.908000
32109	4.910000
32110	4.912000
32111	4.908000
32112	4.903000

Flagged values:

	Wind Speed
18499	4.917000

Showing a flagged example (row 18499) with 5 rows before and 5 rows after (if available) the fla

	Wind Speed
18494	0.069000
18495	0.068000
18496	0.067000
18497	0.067000
18498	0.070000
18499	4.917000
18500	4.916000
18501	4.918000
18502	4.924000
18503	4.922000



Columns(s): general diffuse flows

Issue ID: 2

A strong pattern, and exceptions to the pattern, were found.

Description: The values in "general diffuse flows" are consistently similar to the previous value, they are similar to the median value of the column (5.0355), with exceptions.

Number of exceptions: 160 (0.3053% of rows)

Examples of values NOT flagged (showing a consecutive set of rows):

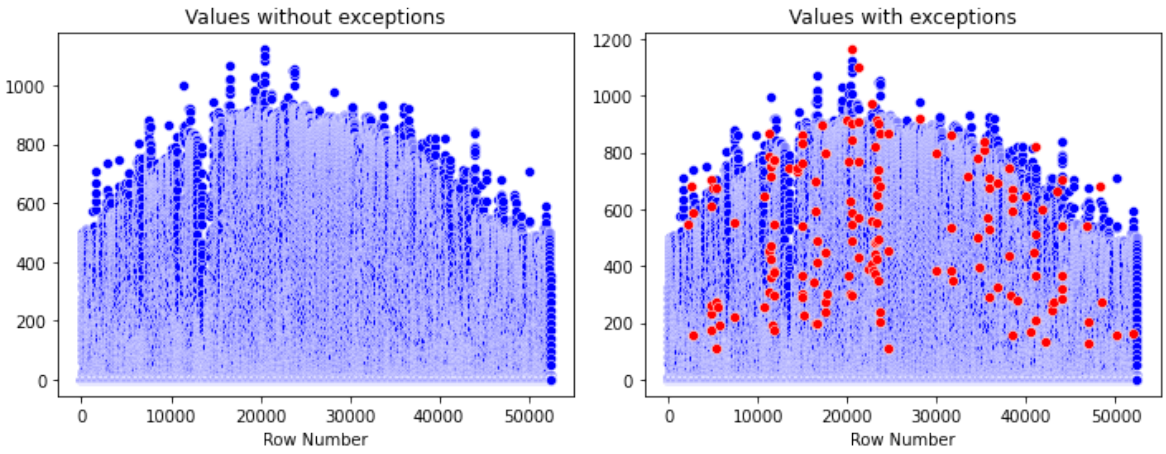
	general diffuse flows
41993	186.000000
41994	204.600000
41995	133.300000
41996	130.200000
41997	147.900000
41998	106.400000
41999	108.700000
42000	112.400000
42001	125.600000
42002	235.600000

Examples of flagged values:

	general diffuse flows
2100	547.100000
2529	679.100000
2672	585.200000
2814	157.800000
4692	702.000000
4695	233.100000
4818	259.300000
4832	613.800000
4837	172.200000
4976	674.200000

Showing a flagged example (row 2100) with 5 rows before and 5 rows after (if available) the flag

	general diffuse flows
2095	491.600000
2096	513.200000
2097	428.400000
2098	311.200000
2099	206.200000
2100	547.100000
2101	530.700000
2102	441.000000
2103	461.600000
2104	455.900000



Columns(s): Zone 1

Issue ID: 3

A strong pattern, and exceptions to the pattern, were found.

Description: The values in "Zone 1" are consistently similar to the previous value, more so than to the median value of the column (32265.92034), with exceptions.

Number of exceptions: 1 (0.0019% of rows)

Examples of values NOT flagged (showing a consecutive set of rows):

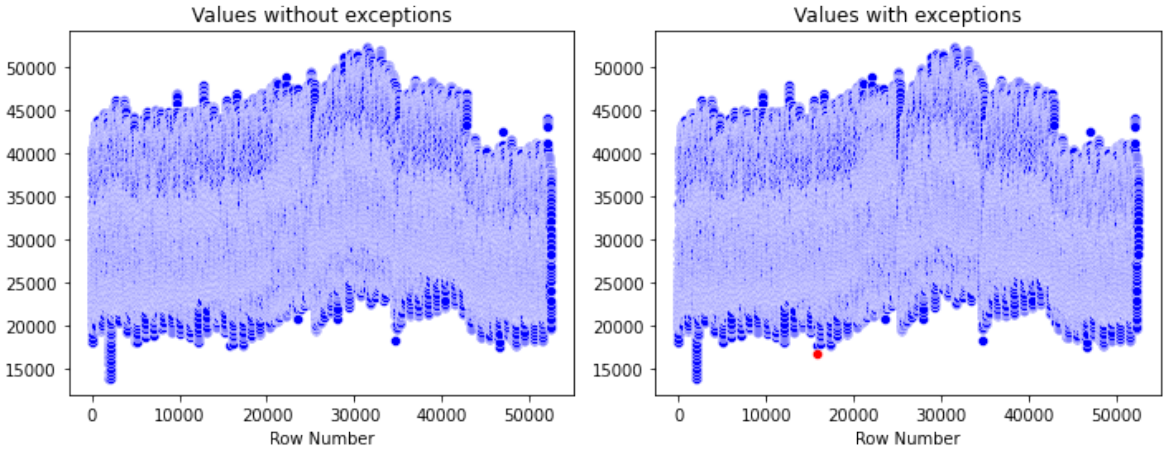
	Zone 1
20757	26137.180330
20758	25835.016390
20759	25457.311480
20760	25432.131150
20761	25344.000000
20762	24953.704920
20763	24500.459020
20764	23568.786890
20765	22511.213110
20766	21856.524590

Flagged values:

	Zone 1
15769	16814.983850

Showing a flagged example (row 15769) with 5 rows before and 5 rows after (if available) the fla

	Zone 1
15764	35824.843920
15765	36097.653390
15766	36544.068890
15767	37015.285250
15768	36159.655540
15769	16814.983850
15770	17794.617870
15771	26995.737350
15772	26822.131320
15773	26226.910660



Columns(s): Zone 2

Issue ID: 4

A strong pattern, and exceptions to the pattern, were found.

Description: The values in "Zone 2 " are consistently similar to the previous value, more so thar similar to the median value of the column (20823.168404999997), with exceptions.

Number of exceptions: 1 (0.0019% of rows)

Examples of values NOT flagged (showing a consecutive set of rows):

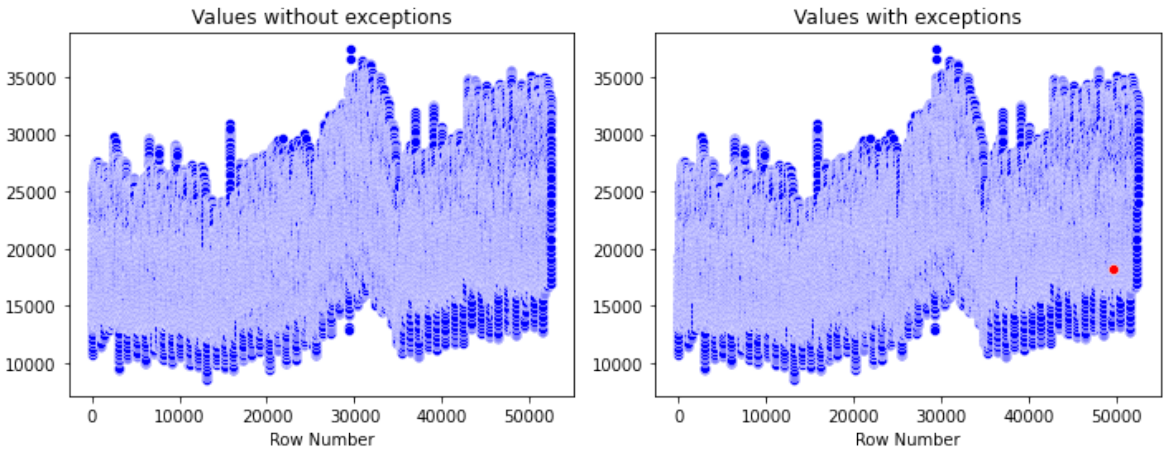
	Zone 2
46884	24809.504130
46885	25077.272730
46886	24723.966940
46887	23347.933880
46888	23704.958680
46889	24422.727270
46890	25073.553720
46891	24861.570250
46892	24686.776860
46893	24207.024790

Flagged values:

	Zone 2
49602	18244.860390

Showing a flagged example (row 49602) with 5 rows before and 5 rows after (if available) the fla

	Zone 2
49597	24375.575330
49598	24710.647440
49599	24957.348880
49600	25656.949980
49601	26058.300090
49602	18244.860390
49603	22781.221230
49604	25704.817430
49605	25940.472540
49606	26260.816200



2 Question 2

Suppose there exists a pair of relations R1(W,X,Y) and R2(X,Y,Z) having t1 $>$ 0 and t2 $>$ 0 tuples, respectively. Consider that W, X, Y and Z take integer values only. Without making any further assumptions, find out the minimum and maximum possible number of tuples that may appear in the resulting relations provided by the following operations.

(i) $(R_1 \cup R_1) \bowtie (R_2 \cap R_2)$

$R_1 \cup R_1 = R_1$ and $R_2 \cap R_2 = R_2$. Therefore, minimum and maximum number of tuples $= R_1 \bowtie R_2$.
Minimum: 0 (when no tuples from R_1 match tuples from R_2 on the common attributes X, Y)
Maximum: $t_1 \times t_2$ (when all tuples in R_1 match all tuples in R_2, i.e., X, Y values are identical across both relations)

(ii) $\pi_{X,Y}(R_1) - \pi_{X,Y}(R_2)$

Minimum: 0 (when $\pi_{X,Y}(R_1) \subseteq \pi_{X,Y}(R_2)$, so the difference is empty)
Maximum: t_1 (when there is no overlap between $\pi_{X,Y}(R_1)$ and $\pi_{X,Y}(R_2)$, so all tuples in $\pi_{X,Y}(R_1)$ remain)

(iii) $(R_1 - R_2) \bowtie (R_2 - R_1)$

The attribute set of the relations is unequal therefore $(R_1 - R_2) and (R_2 - R_1)$ is invalid.
Minimum: invalid
Maximum: invalid

(iv) $R_1 \div (\pi_{X,Y}(R_1) \cap \pi_{X,Y}(R_2))$

Minimum: 0 (If none of the distinct W-values in R_1 are associated with every tuple in the common part $\pi_{X,Y}(R_1) \cap \pi_{X,Y}(R_2)$, the result is empty.)
Maximum: t_1 (If all distinct W-values in R_1 are associated with every tuple in the common part $\pi_{X,Y}(R_1) \cap \pi_{X,Y}(R_2)$, all W-values are included in the result.)