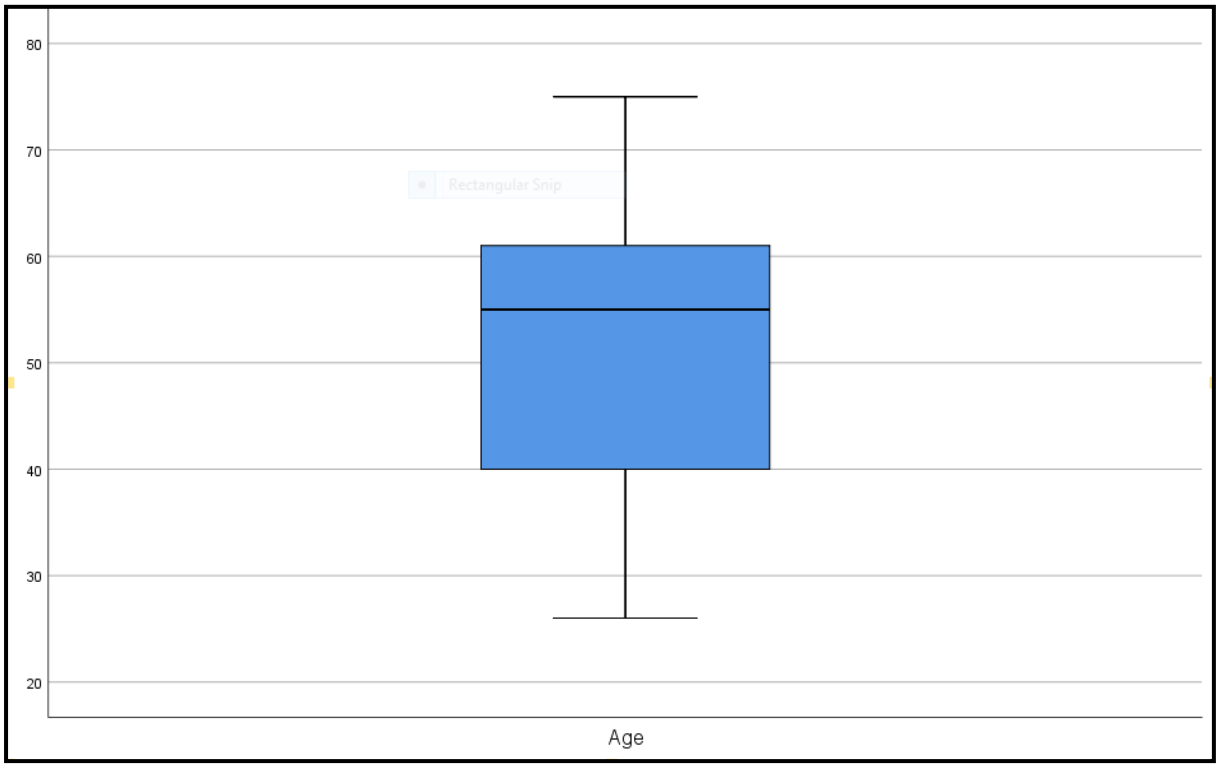
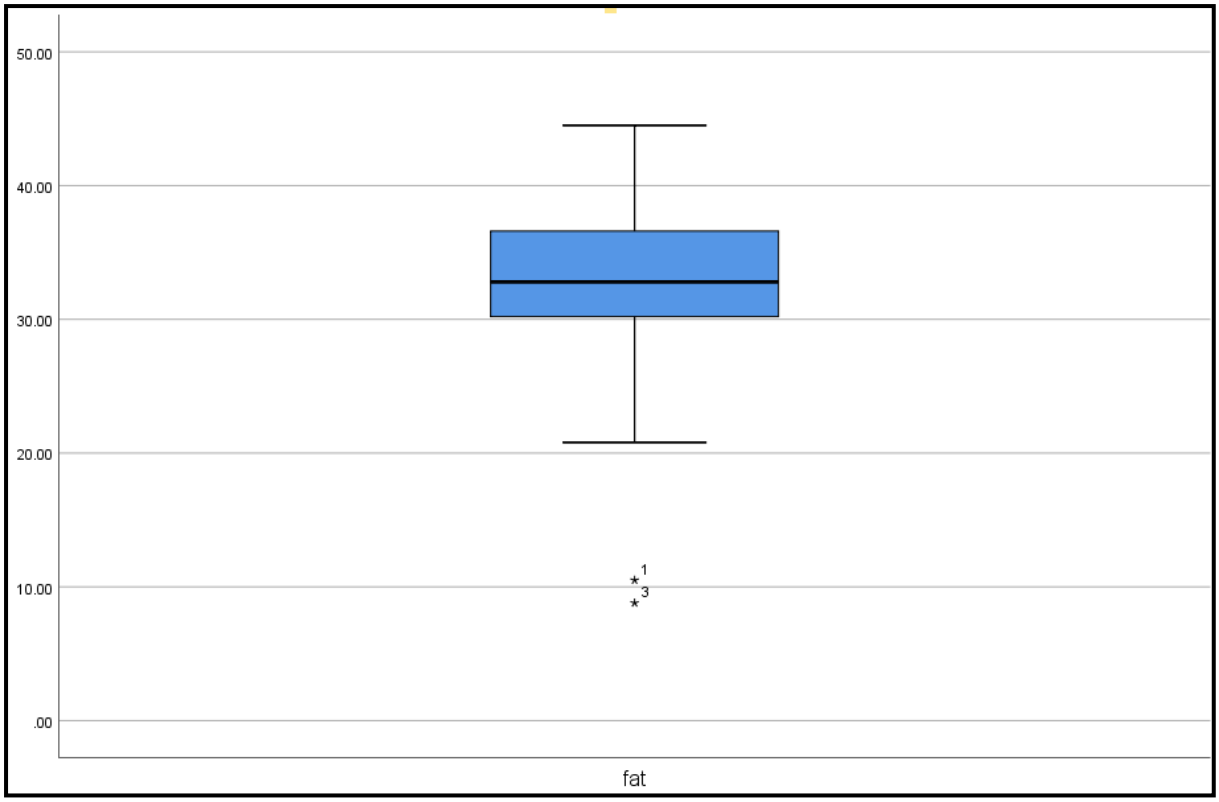
Section a)

Boxplot Age



Conclusion: According to boxplot, the minimum age is 27 while the maximum age is 85. The first and third quartile is between 40 and 61. The median value is at 55.
There are no outliers in this variable.

Boxplot %fat



Conclusion: According to boxplot, the minimum %fat is 21.00 while the maximum %fat is 44.00. The first and third quartile is between 30.00 and 47.00. The median value is at 33.0.
There are two outliers in this data. Both the outliers are near 10 and highlighted with a star (*).

Section b)

In ***z*-score normalization** (or *zero-mean normalization*), the values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, *vi* , of *A* is normalized to *v*0*i* by computing

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A},$$

where N*A* and _*A* are the mean and standard deviation, respectively, of attribute *A*.

This method of normalization is useful when the actual minimum and maximum of attribute *A* are unknown.

Age	%fat	ZAge	Z%fat
26	10.5	-1.61828	-2.1553
26	30.5	-1.61828	-0.05882
29	8.8	-1.41693	-2.33351
29	20.8	-1.41693	-1.07561
40	32.4	-0.67863	0.14035
45	26.9	-0.34305	-0.43619
50	30.4	-0.00746	-0.0693

55	30.2	0.32813	-0.09027
60	33.2	0.66372	0.22421
55	36.6	0.32813	0.58061
45	44.5	-0.34305	1.40872
60	30.8	0.66372	-0.02737
55	35.4	0.32813	0.45482
61	33.2	0.73084	0.22421
62	36.1	0.79795	0.5282
63	37.9	0.86507	0.71688
75	43.2	1.67048	1.27245
66	37.7	1.06642	0.69592

Section c)

In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such an attribute greater effect or "weight." To help avoid dependence on the choice of measurement units, the data should be *normalized* or *standardized*. This involves transforming the data to fall within a smaller or common range such as [-1, 1] or [0.0, 1.0]. Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbour classification and clustering.

i)

Min-max normalization: Min-max normalization performs a linear transformation on the original data. Suppose that *minA* and *maxA* are the minimum and maximum values of an attribute, *A*. Min-max normalization maps a value, *vi* , of *A* to *v*0*i* in the range [*new minA*, *new maxA*] by computing

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

Min-max normalization preserves the relationships among the original data values. It will encounter an "out-of-bounds" error if a future input case for normalization falls outside of the original data range for *A*.

Example:

Row 3 (Highlighted in Yellow)
Age: 29
Maximum Age: 75
Minimum Age: 26

MMAge: ((29-26)/(75-26)) * (1.0 - 0.0) + (0.0) = 0.06122

	Age	%fat	MMAge	MM%fat
	26	10.5	0	0.047619
	26	30.5	0	0.607843
	29	8.8	0.061224	0

	29	20.8	0.061224	0.336134
	40	32.4	0.285714	0.661064
	45	26.9	0.387755	0.507003
	50	30.4	0.489796	0.605042
	55	30.2	0.591837	0.59944
	60	33.2	0.693878	0.683473
	55	36.6	0.591837	0.778711
	45	44.5	0.387755	1
	60	30.8	0.693878	0.616246
	55	35.4	0.591837	0.745098
	61	33.2	0.714286	0.683473
	62	36.1	0.734694	0.764706
	63	37.9	0.755102	0.815126
	75	43.2	1	0.963585
	66	37.7	0.816327	0.809524
Max	75	44.5		
Min	26	8.8		

ii)

***z*-score normalization**: In ***z*-score normalization** (or *zero-mean normalization*), the values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, *vi* , of *A* is normalized to *v*0*i* by computing

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A},$$

where N*A* and _*A* are the mean and standard deviation, respectively, of attribute *A*.

This method of normalization is useful when the actual minimum and maximum of attribute *A* are unknown.

Example:

Row 5 (Highlighted in Yellow)
Age: 40
Mean Age: 50.11
Standard Deviation Age: 14.9

MMAge: (40 – 50.11)/14.9 = -0.678

	Age	**%fat**	**MMAge**	**MM%fat**
	26	10.5	-1.61828	-2.1553
	26	30.5	-1.61828	-0.05882
	29	8.8	-1.41693	-2.33351
	29	20.8	-1.41693	-1.07561

	40	32.4	-0.67863	0.14035
	45	26.9	-0.34305	-0.43619
	50	30.4	-0.00746	-0.0693
	55	30.2	0.32813	-0.09027
	60	33.2	0.66372	0.22421
	55	36.6	0.32813	0.58061
	45	44.5	-0.34305	1.40872
	60	30.8	0.66372	-0.02737
	55	35.4	0.32813	0.45482
	61	33.2	0.73084	0.22421
	62	36.1	0.79795	0.5282
	63	37.9	0.86507	0.71688
	75	43.2	1.67048	1.27245
	66	37.7	1.06642	0.69592
Mean	50.11111	31.06111		
StDeviation	14.89923	9.539771		

iii)

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute *A*. The number of decimal points moved depends on the maximum absolute value of *A*. A value, *vi* , of *A* is normalized to *v0i* by computing

$$v_i' = \frac{v_i}{10^j},$$

where *j* is the smallest integer such that $max(|v_i'|) < 1.$

Example:

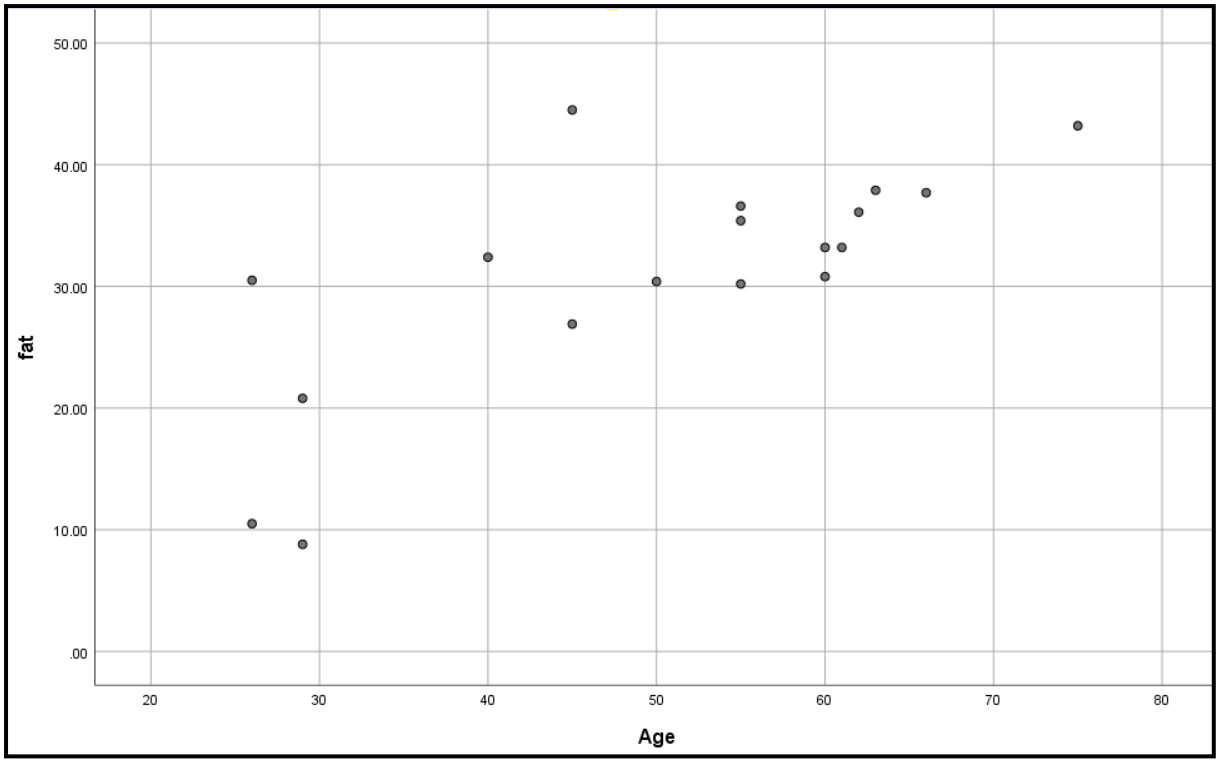
Row 7 (Highlighted in Yellow)
Age: 50
Maximum Absolute Age: 75

Scaling by Decimal (SDAge): 50/100 = 0.5

	Age	**%fat**	**SDAge**	**SD%fat**
	26	10.5	0.26	0.105
	26	30.5	0.26	0.305
	29	8.8	0.29	0.088
	29	20.8	0.29	0.208
	40	32.4	0.4	0.324
	45	26.9	0.45	0.269
	50	30.4	0.5	0.304
	55	30.2	0.55	0.302
	60	33.2	0.6	0.332

	55	36.6	0.55	0.366
	45	44.5	0.45	0.445
	60	30.8	0.6	0.308
	55	35.4	0.55	0.354
	61	33.2	0.61	0.332
	62	36.1	0.62	0.361
	63	37.9	0.63	0.379
	75	43.2	0.75	0.432
	66	37.7	0.66	0.377
Range	26 - 75	8.8 - 44.5		

Section d)



Conclusion: The two variables Age and %fat show a linear relationship with each other. As the age of a person increases, the % fat also increases. There is a positive linear and strong relationship between the two variables.
The minimum age of a person is around late 20's while the % fat is between 10.00 to 30.00.
The maximum age of the person is 75 and the % fat is ~ 43.

Section e)

Correlation is a measurement of how strong are two variables linearly related. **Correlation coefficient** is a number **between** -1 and 1 that shows the result of **correlation**. The closer it is to 1, the stronger positive linear **relationship** do the two variables have.

Covariance measures how two variables move with respect to each other and is an extension of the concept of variance (which tells about how a single variable varies). The values lie between -∞ and +∞.

Correlations

		Age	fat
Age	Pearson Correlation	1	.735**
	Sig. (2-tailed)		.001
	N	18	18
fat	Pearson Correlation	.735**	1
	Sig. (2-tailed)	.001	
	N	18	18

**. Correlation is significant at the 0.01 level (2-tailed).

The variables age and % fat have a positive correlation with each other at 0.735. According to two tailed test, it is below the level of significance at 0.001.

Covariance

		Age	fat
Age	Pearson Correlation	1	.735**
	Sig. (2-tailed)		.001
	Sum of Squares and Cross-products	3773.778	1776.578
	Covariance	221.987	104.505
	N	18	18
fat	Pearson Correlation	.735**	1
	Sig. (2-tailed)	.001	
	Sum of Squares and Cross-products	1776.578	1547.123
	Covariance	104.505	91.007
	N	18	18

**. Correlation is significant at the 0.01 level (2-tailed).

The variance between age and % fat is 104.505. According to two tailed test, it is below the level of significance at 0.001.

Problem 2)

Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering.
Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it. The sorted values are distributed into a number of "buckets," or *bins*. Because binning methods consult the neighborhood of values, they perform *local* smoothing.

In **smoothing by bin means**, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9. Similarly, **smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median. In **smoothing by bin boundaries**, the minimum and maximum values in a given bin are identified as the *bin boundaries*. Each bin value is then replaced by the closest boundary value.

In general, the larger the width, the greater the effect of the smoothing.
Alternatively, bins may be *equal width*, where the interval range of values in each bin is constant. Binning is also used as a discretization technique.

Problem 2

8, 13, 14, 15, 17, 37, 55, 60, 77, 95, 208, 218

Section a)

Equal depth partitioning

Bin 1 : 8, 13, 14
Bin 2 : 15, 17, 37
Bin 3 : 55, 60, 77
Bin 4 : 95, 208, 218

Smoothing by means

Bin 1 : 11.67, 11.67, 11.67
Bin 2 : 23, 23, 23
Bin 3 : 64, 64, 64
Bin 4 : 173.67, 173.67, 173.67

Smoothing by median

Bin 1 : 13, 13, 13
Bin 2 : 17, 17, 17
Bin 3 : 60, 60, 60
Bin 4 : 208, 208, 208

Smoothing by boundaries
Bin 1 : 8, 14, 14
Bin 2 : 15, 15, 37
Bin 3 : 55, 55, 77
Bin 4 : 95, 218, 218

Section b)

Equal width partitioning

$$\text{width} = \frac{\text{max} - \text{min}}{\text{Number of bins}} = \frac{218 - 8}{4} = \frac{210}{4} = 52.5$$

$52.5 + 8 = 60.5$ (from 8 to 60.5)
Bin 1 : 8, 13, 14, 15, 17, 37, 55, 60

$60.5 + 52.5 = 113$ (from 60.5 to 113)
Bin 2 : 77, 95

$113 + 52.5 = 165.5$ (from 113 to 165.5)
Bin 3 :

$165.5 + 52.5 = 218$ (from 165.5 to 218)
Bin 4 : 208, 218

Problem 3)

Section a)

Since the employed/unemployed variable has distinctive grouping and clustering which can help to classify data based on either circle or plus class. We would use the clustering technique which partitions the objects into groups or clusters so that objects within a cluster are similar to one another and dissimilar objects in the other clusters.

Section b)

An **attribute selection measure** is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of class-labeled training tuples into individual classes.
The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure4 is chosen as the *splitting attribute* for the given tuples. If the splitting attribute is continuous-valued or if we are restricted to binary trees, then, respectively, either a *split point* or a *splitting subset* must also be determined as part of the splitting criterion. The tree node created for partition *D* is labeled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly.

ID3 uses **information gain** as its attribute selection measure.
Let node *N* represent or hold the tuples of partition *D*. The attribute with the highest information gain is chosen as the splitting attribute for node *N*. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions.
Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.
The expected information needed to classify a tuple in *D* is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

where *pi* is the nonzero probability that an arbitrary tuple in *D* belongs to class *Ci* and is estimated by $|C_{i,D}|/|D|$. A log function to the base 2 is used, because the information is encoded in bits. *Info*(*D*) is just the average amount of information needed to identify the class label of a tuple in *D*.

Info(*D*) is also known as the **entropy** of *D*.
Now, suppose we were to partition the tuples in *D* on some attribute *A* having *v* distinct values, {a1, a2, ... an}, as observed from the training data. If *A* is discrete-valued, these values correspond directly to the *v* outcomes of a test on *A*. Attribute *A* can be used to split *D* into *v* partitions or subsets, {D1, D2, … Dn} where *Dj* contains those tuples in *D* that have outcome *aj* of *A*. These partitions would correspond to the branches grown from node *N*.
This amount is measured by

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

InfoA.*D*/ is the expected information required to classify a tuple from *D* based on the partitioning by *A*. The smaller the expected information (still) required, the greater the purity of the partitions.

In other words, *Gain*(*A*) tells us how much would be gained by branching on *A*. It is the expected reduction in the information requirement caused by knowing the value of *A*. The attribute *A* with the highest information gain, *Gain*. *A*/, is chosen as the splitting attribute at node *N*.

Steps:
The class label attribute, *C*, has two distinct values (namely, I, II)
There are two tuples of class *I* and two tuples of class *II*. A (root) node *N* is created for the tuples in Problem 3 (D).

To find the splitting criterion for these tuples, we must compute the information gain of each attribute.

Info(D) = - (2/4) log2(2/4) – (2/4) log2(2/4) = - 0.5 * (-1) – (0.5) * (-1) = 1

We need to compute the expected information requirement for each attribute.
Let's start with the attribute *X*
We need to look at the distribution of *I* and *II* tuples for each category of *X*.

For the *X* category "1," there are two *I* tuples and 1 *II* tuples.
For the X category "0," there are 1 *II* tuples and zero *I* tuples.

Info X (D) = (3/4) * (- (2/3) log2(2/3) - (1/3) log2(1/3)) + (1/4) * (- (1/1) log2(1/1))
= (3/4) * ((- 2/3) * (- 0.57) – (1/3) * (- 1.6)) + (0)
= (3/4) * (0.38 + 0.48)
= 0.645

Hence, the gain in information from such a partitioning would be
Gain (x) = Info(D) - Info X (D)
= 1 – 0.645
= 0.355 bits

Similarly we find the value for attribute *Y.*

For the *Y* category "1," there are two *I* tuples and zero *II* tuples.
For the Y category "0," there are two *II* tuples and zero *I* tuples.

Info y (D) = (2/4) * (- (2/2) log2(2/2)) + (2/4) * (- (2/2) log2(2/2))
= (0.5) * (0) + (0.5) * (0)
= 0

The gain in information from such a partitioning would be

Gain (y) = Info(D) - Info y (D)
= 1 – 0
= 0 bits

The value for attribute *Z.*

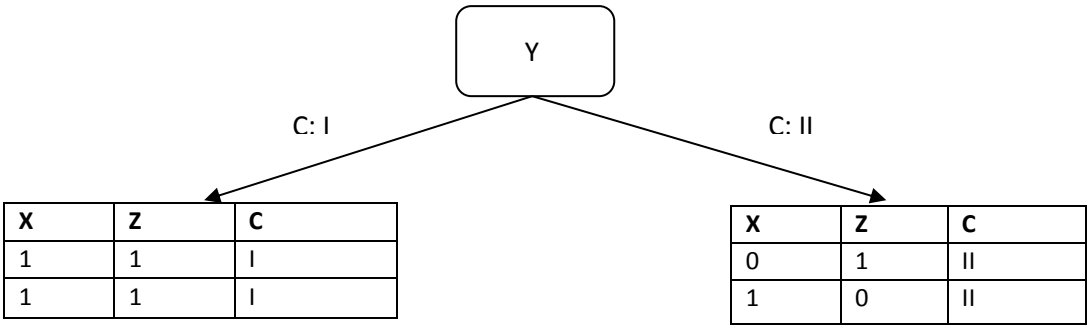
For the *Z* category "1," there are two *I* tuples and 1 *II* tuples.
For the Z category "0," there are 1 *II* tuples and zero *I* tuples

Info X (D) = (3/4) * (- (2/3) log2(2/3) - (1/3) log2(1/3)) + (1/4) * (- (1/1) log2(1/1))
= (3/4) * ((- 2/3) * (- 0.57) – (1/3) * (- 1.6)) + (0)
= (3/4) * (0.38 + 0.48)
= 0.645

Hence, the gain in information from such a partitioning would be

Gain (z) = Info(D) - Info z (D)
= 1 – 0.645
= 0.355 bits

Because *Y* has the highest information gain among the attributes, it is selected as the splitting attribute. Node *N* is labeled with *Y*, and branches are grown for each of the attribute's values. The tuples are then partitioned accordingly.



X	Z	C
1	1	I
1	1	I

X	Z	C
0	1	II
1	0	II

Problem 4

Section a)

There are total of 19 attributes in the spotify dataset.

Attribute Name	Type	Measure
index	Numeric	Scale
id	String	Nominal
name	String	Nominal
uri	String	Nominal
artist	String	Nominal
acousticness	Scientific	Scale
danceability	Numeric	Scale
duration_ms	Numeric	Scale
energy	Numeric	Scale
instrumentalness	Scientific	Scale
key	Numeric	Nominal
liveness	Numeric	Scale
loudness	Numeric	Scale
mode	Numeric	Nominal
speechiness	Numeric	Scale
tempo	Numeric	Scale
time_signature	Numeric	Nominal
valence	Scientific	Scale
moods	String	Nominal

Statistics

		key	mode	time_signature	index	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
N	Valid	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420
	Missing	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean		5.14	0.59	3.89	709.84	3.996E-01	0.556092	283411.48	0.5546822	2.805E-01	0.191559	-10.73853	0.082647	116.85777	3.982E-01
Median		5.00	1.00	4.00	709.50	2.650E-01	0.588000	226800.00	0.5965000	2.075E-03	0.121000	-7.81100	0.048750	118.01650	3.645E-01
Std. Deviation		3.566	0.492	0.484	410.453	3.8140E-01	0.1894721	337189.066	0.29405561	3.9252E-01	0.1657403	7.986422	0.0828924	27.948877	2.5826E-01
Minimum		0	0	1	0	2.3E-05	0.0585	54333	0.00154	0.0E+00	0.0227	-41.808	0.0229	52.799	1.0E-05
Maximum		11	1	5	1420	1.0E+00	0.9670	4500037	1.00000	1.0E+00	0.9790	-0.750	0.5200	213.973	9.7E-01
Percentiles	25	2.00	0.00	4.00	354.25	3.138E-02	0.447250	196043.75	0.3172500	1.023E-06	0.096625	-14.41000	0.037000	95.04975	1.863E-01
	50	5.00	1.00	4.00	709.50	2.650E-01	0.588000	226800.00	0.5965000	2.075E-03	0.121000	-7.81100	0.048750	118.01650	3.645E-01
	75	8.00	1.00	4.00	1065.75	8.150E-01	0.691000	274670.25	0.8010000	7.985E-01	0.231750	-5.03350	0.081475	131.50450	5.880E-01

The above tables summarises the mean, median, minimum, maximum and the quartiles for each attribute.

Example:
Key column
The mean for this column is 5.14 while the median is 5.
The standard deviation for this column is marked at 3.566.
The first and third quartile is between 2 and 8.
The minimum value is 0 and the maximum value is 11.

Number of tuples: 1420

Number of Cases and distribution

key

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	154	10.8	10.8	10.8
	1	181	12.7	12.7	23.6
	2	126	8.9	8.9	32.5
	3	58	4.1	4.1	36.5
	4	104	7.3	7.3	43.9
	5	142	10.0	10.0	53.9
	6	86	6.1	6.1	59.9
	7	146	10.3	10.3	70.2
	8	104	7.3	7.3	77.5
	9	113	8.0	8.0	85.5
	10	89	6.3	6.3	91.8
	11	117	8.2	8.2	100.0
	Total	1420	100.0	100.0	

The Key attribute column has value ranges from 0 – 11. The frequency column in the above screenshot represents the number of cases for each value.

mode

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	585	41.2	41.2	41.2
	1	835	58.8	58.8	100.0
	Total	1420	100.0	100.0	

The Mode attribute column has two values 1 and 0. There are 585 records for value 0 while 835 records for value 1.

time_signature

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	21	1.5	1.5	1.5
	3	126	8.9	8.9	10.4
	4	1239	87.3	87.3	97.6
	5	34	2.4	2.4	100.0
	Total	1420	100.0	100.0	

The Time Signature attribute column has values 1, 3, 4 and 5. The value 4 has maximum number of cases at 1239.

moods

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	dinner	467	32.9	32.9	32.9
	dinner, party	3	.2	.2	33.1
	dinner, workout	1	.1	.1	33.2
	party	225	15.8	15.8	49.0
	party, workout	52	3.7	3.7	52.7
	sleep	362	25.5	25.5	78.2
	workout	310	21.8	21.8	100.0
	Total	1420	100.0	100.0	

The moods column has 7 unique values. Some records have only one value (dinner) while some records hold a combination of values (dinner, party).
The category dinner has maximum number of records at 467 while dinner and workout has minimum record at 1.
The moods column has no value missing with 1420 records split among all 7 unique values.

Correlations:
Correlation is a measurement of how strong are two variables linearly related. **Correlation coefficient** is a number **between** -1 and 1 that shows the result of **correlation**. The closer it is to 1, the stronger positive linear **relationship** do the two variables have.

Correlations													
	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence
acousticness	1	-.526**	.056*	-.816**	.566**	-0.042	-.217**	-.724**	.077**	-.319**	-.220**	-.254**	-.365**
danceability	-.526**	1	-.302**	.436**	-.569**	0.031	-.105**	.652**	-.067*	.208**	.146**	.296**	.627**
duration_ms	.056*	-.302**	1	0.046	.155**	-.071**	.180**	-.203**	0.042	-0.014	-.119**	-.076**	-.216**
energy	-.816**	.436**	0.046	1	-.538**	0.045	.332**	.777**	-.055*	.282**	.211**	.238**	.400**
instrumentalness	.566**	-.569**	.155**	-.538**	1	-0.014	-.062*	-.726**	-0.026	-.263**	-.173**	-.261**	-.505**
key	-0.042	0.031	-.071**	0.045	-0.014	1	0.033	0.021	-.178**	.088**	-0.044	0.021	.083**
liveness	-.217**	-.105**	.180**	.332**	-.062*	0.033	1	.111**	-0.018	.128**	0.014	0.023	-.067*
loudness	-.724**	.652**	-.203**	.777**	-.726**	0.021	.111**	1	-0.034	.252**	.262**	.299**	.488**
mode	.077**	-.067*	0.042	-.055*	-0.026	-.178**	-0.018	-0.034	1	-.081**	-0.015	-0.008	-.064*
speechiness	-.319**	.208**	-0.014	.282**	-.263**	.088**	.128**	.252**	-.081**	1	.145**	.122**	.150**
tempo	-.220**	.146**	-.119**	.211**	-.173**	-0.044	0.014	.262**	-0.015	.145**	1	.054*	.094**
time_signature	-.254**	.296**	-.076**	.238**	-.261**	0.021	0.023	.299**	-0.008	.122**	.054*	1	.180**
valence	-.365**	.627**	-.216**	.400**	-.505**	.083**	-.067*	.488**	-.064*	.150**	.094**	.180**	1

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

The maximum negative correlation is between acousticness and energy at -0.816.
The maximum positive correlation is between loudness and energy at 0.777.

There are high chances of multicollinearity between the variables in this dataset.
Variables like instrumentalness, energy, loudness and valence have significant correlation with other variables.

Section b)

Statistics														
		key	mode	time_signature	index	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence
N	Valid	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420	1420
	Missing	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mean		5.14	0.59	3.89	709.84	3.996E-01	0.556092	283411.48	0.5546822	2.805E-01	0.191559	-10.73853	0.082647	116.85777	3.982E-01
Median		5.00	1.00	4.00	709.50	2.650E-01	0.588000	226800.00	0.5965000	2.075E-03	0.121000	-7.81100	0.048750	118.01650	3.645E-01
Std. Deviation		3.566	0.492	0.484	410.453	3.8140E-01	0.1894721	337189.066	0.29405561	3.9252E-01	0.1657403	7.986422	0.0828924	27.948877	2.5826E-01
Minimum		0	0	1	0	2.3E-05	0.0585	54333	0.00154	0.0E+00	0.0227	-41.808	0.0229	52.799	1.0E-05
Maximum		11	1	5	1420	1.0E+00	0.9670	4500037	1.00000	1.0E+00	0.9790	-0.750	0.5200	213.973	9.7E-01
Percentiles	25	2.00	0.00	4.00	354.25	3.138E-02	0.447250	196043.75	0.3172500	1.023E-06	0.096625	-14.41000	0.037000	95.04975	1.863E-01
	50	5.00	1.00	4.00	709.50	2.650E-01	0.588000	226800.00	0.5965000	2.075E-03	0.121000	-7.81100	0.048750	118.01650	3.645E-01
	75	8.00	1.00	4.00	1065.75	8.150E-01	0.691000	274670.25	0.8010000	7.985E-01	0.231750	-5.03350	0.081475	131.50450	5.880E-01

The Minimum Maximum value from the above summary report can help us find the range for each variable.

Attribute	Lower Range	Upper Range
danceability	0.0585	0.9670
duration_ms	54333	4500037
energy	0.00154	1
instrumentalness	0	1
key	0	11
liveness	0.0227	0.9790
loudness	-41.808	-0.750
mode	0	1
speechiness	0.0229	0.5200
tempo	52.799	213.973
time_signature	1	5

Since the values of the variables are within the range of {-1, 1} or {0, 1}.
The following variables do not require any normalization technique.

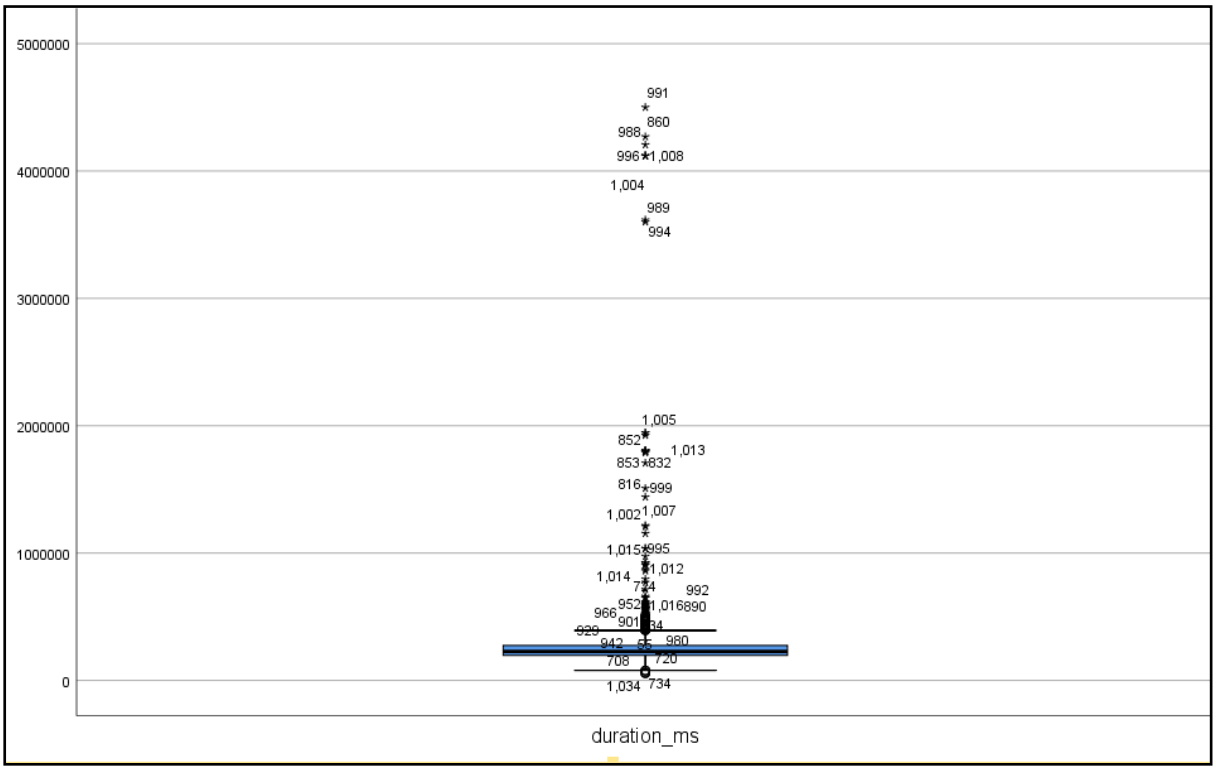
- Danceability
- Energy
- Instrumentalness
- Liveness
- Mode
- speechiness

The Normalisation technique will be applied for the following variables since the variables are having out of range values and outliers.

duration_ms

Range: 54333 – 4500037

Boxplot:

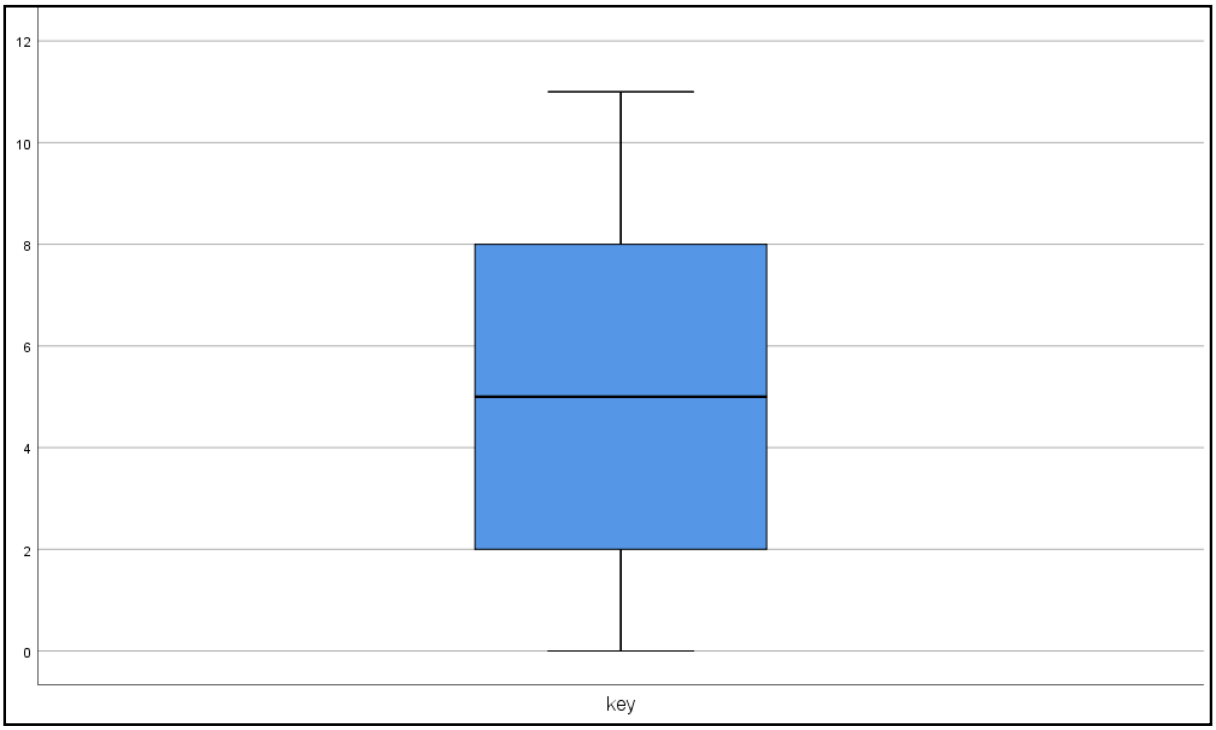


Conclusion: As shown in the above boxplot, the duration_ms variable has many outliers and values are out of range.
We can use the z score normalisation technique.
The outliers would dominate min max normalisation. Hence we are not using the min max normalisation.

key

Range: 0 – 11

Boxplot:

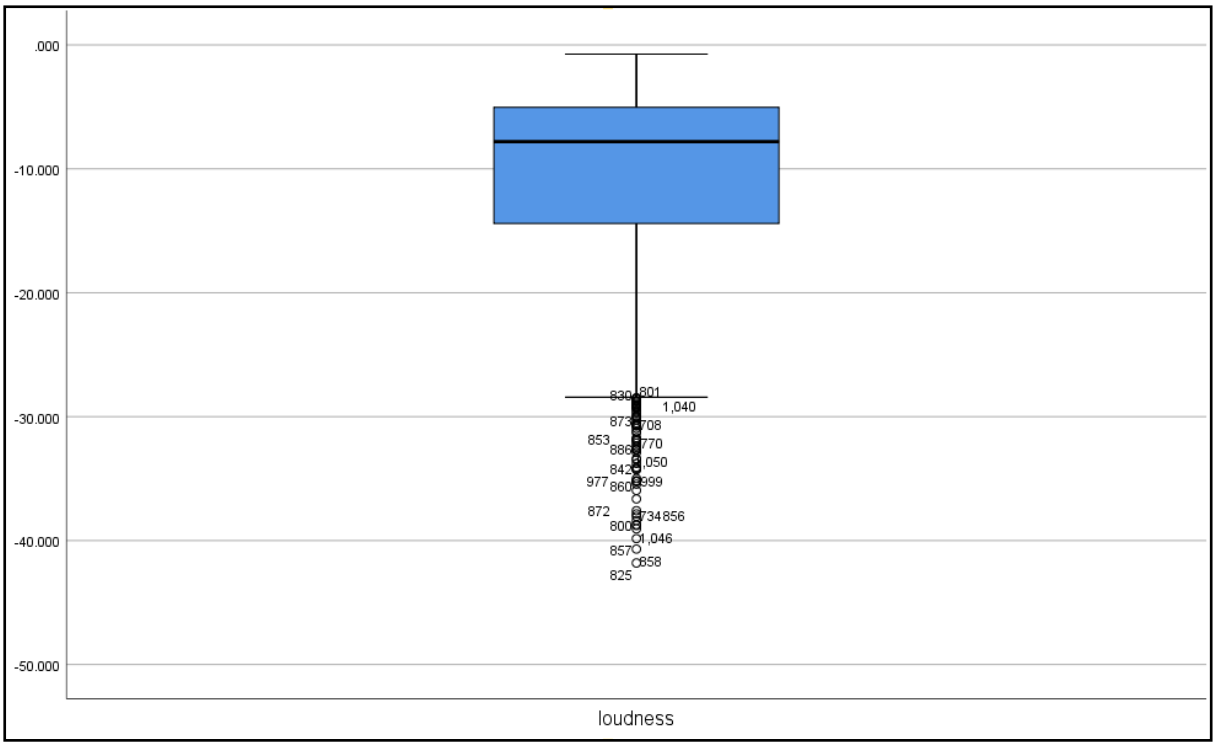


Conclusion: As shown in the above boxplot, the key variable has well defined boxplot. There are no outliers in this plot.
Since there are no outliers in the plot, the values are within certain range and we know the min max value.
We can use the min max score normalisation technique.

loudness

Range: -41.808 - -0.750

Boxplot:

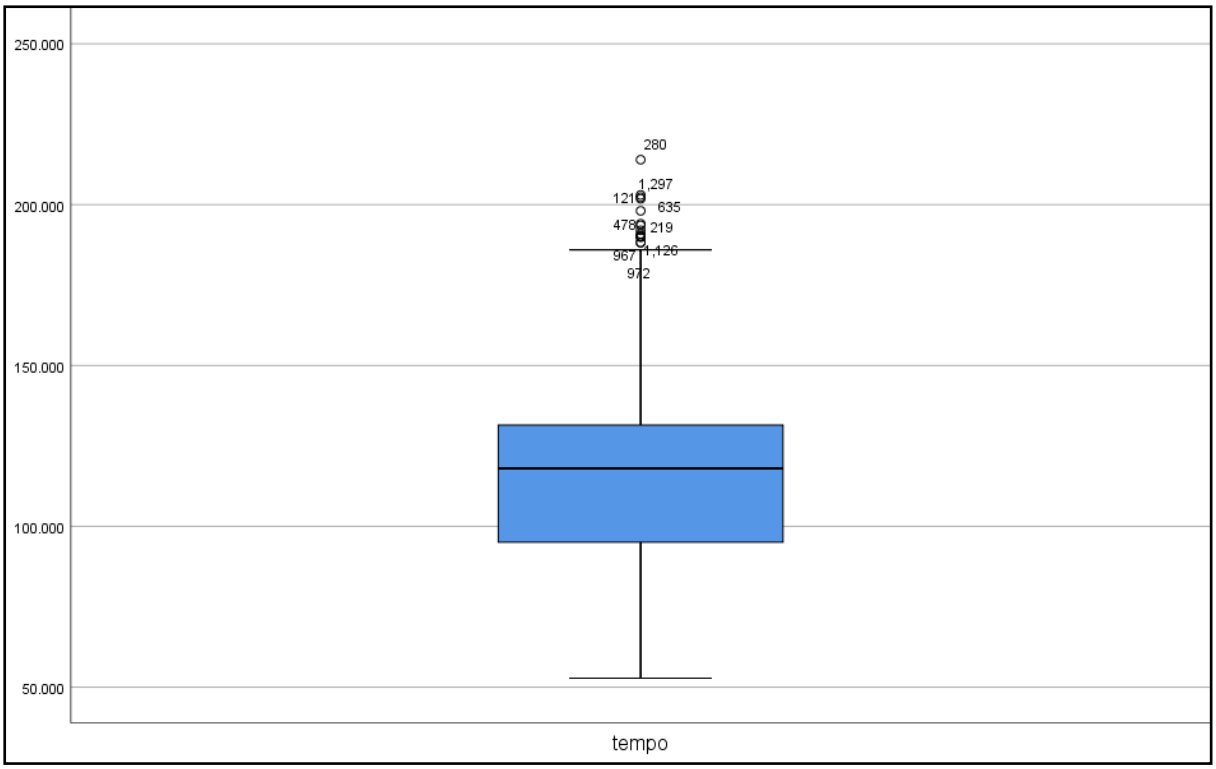


Conclusion: As shown in the above boxplot, the loudness variable has many outliers and values are out of range.
We can use the z score normalisation technique.
The outliers would dominate min max normalisation. Hence we are not using the min max normalisation.

tempo

Range: 52.799 – 213.973

Boxplot:

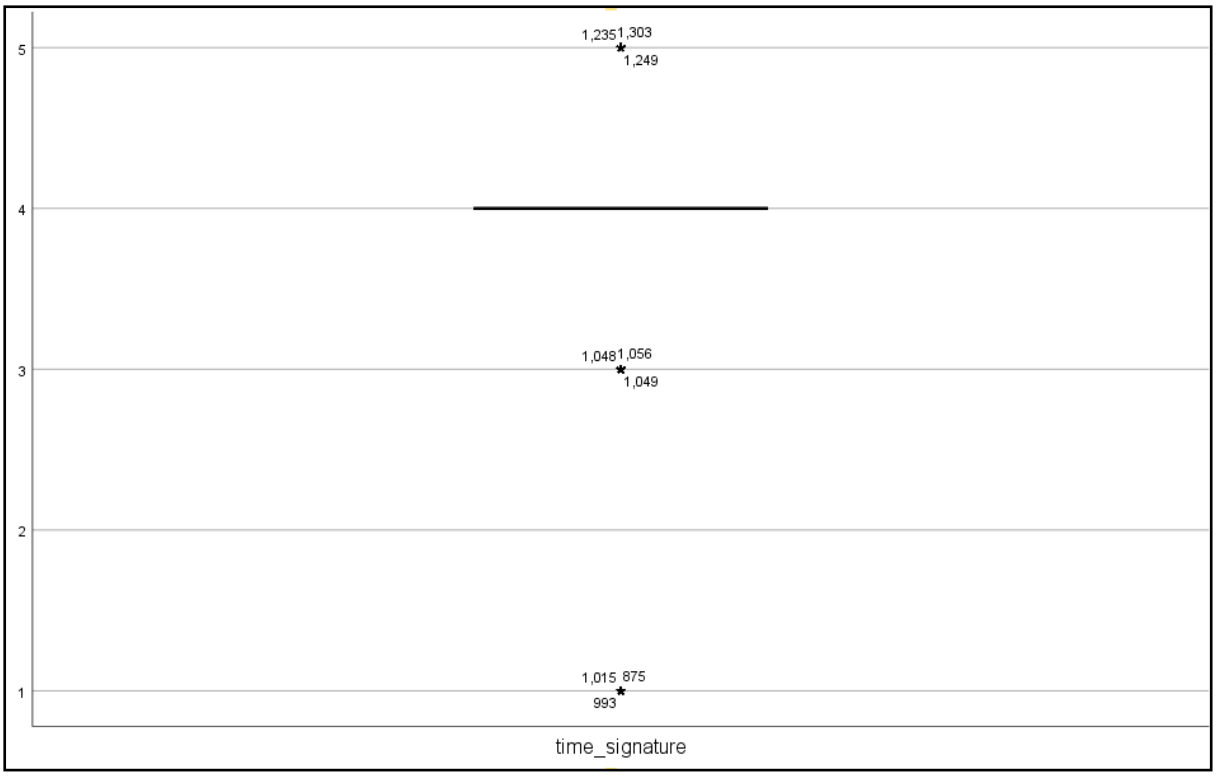


Conclusion: As shown in the above boxplot, the tempo variable has many outliers and values are out of range.
We can use the z score normalisation technique.
The outliers would dominate min max normalisation. Hence we are not using the min max normalisation.

Time_signature

Range: 1 - 5

Boxplot:



Conclusion: As shown in the above boxplot, the time_signature variable has many outliers and values are out of range.
We can use the z score normalisation technique.
The outliers would dominate min max normalisation. Hence we are not using the min max normalisation.