Be Indian – Excel Audit Tool Template – An Introduction

- CA S. Rathinagiri (srathinagiri@gmail.com)

Welcome, this is IT era! Like big bang, today we have big data! Data is rapidly expanding/exploding in various dimensions, thanks to computers, internet, mobile, IoTs and what not?! Development of Hardware, Software, Network and Algorithms made it possible. Starting from a flat file based database, today we are having data centers spreading across the globe. The field of data analytics, data science, artificial intelligence are madly getting innovated by organizations, corporations and individuals.

As an accounting professional, I am so fascinated by the Computer Aided Audit Tools shortly called as CAAT. There are exclusive commercial products like Caseware IDEA and ACL as CAAT. As I use Electronic Spreadsheets a lot it my office, I tried the features of these products in Microsoft Excel products and seen that Excel is having most of them already! What we can get with a click of a button in the Computer Aided Audit Tools, which are exclusively designed for audit purposes, can be done by developing some formulas/macros/functions in Excel.

Until 2016, before the introduction of Power Query and Power Pivot, Excel was struggling when we had data more than 100000 records. Apart from Power Query and Power Pivot the I am seeing two other features as revolutions done by Microsoft Excel in the recent versions especially Excel 365. The first feature is dynamic arrays and functions related to arrays. The second feature is making the users as developers with Let and Lambda functions. Everything blended together, today's Excel is so powerful and capable of doing whatever CAAT can do!

Mantra of data analytics today is "No coding, Low coding". After Python and R, it is well proven that you need not be a developer for analyzing the data. From the day Lambda function was introduced in Excel, I tried to learn about the programming abilities of this function along with other array functions.

This Excel Audit Tool template is collection of 100+ wide range of functions related to financial analysis, statistics, data manipulation, and other Excel operations. I have started this as a hobby project.

When searching for some algorithms and solutions, I have come across various online resources and contributors who have generously shared the algorithms and knowledge related to the functions listed. These resources have been instrumental in the development of these functions, and they have played a significant role in enhancing their utility in various domains, including finance, auditing, data analysis, and more.

The collective wisdom and expertise of the online community have been invaluable that serve both personal and professional purposes. The willingness of individuals and organizations to share their insights and solutions has been instrumental in broadening the scope and capabilities of these functions.

I have named this tool '**Be Indian**' because some of these functions are created within the context of the Indian environment. Furthermore, I have used this name since 2000 for my Excel add-in, which was designed to spell Indian numbers and format numbers using the Indian method of commas.

This template is and will be 100% free and it certainly comes with no warranties or guarantees! If you use this template you are most welcome to give suggestions/comments/critics/ideas for further improvement.

I extend my thanks to the countless individuals, forums, websites, and open-source communities who have contributed to the development of these algorithms. Their dedication to knowledge sharing has enriched me in this learning and development process.

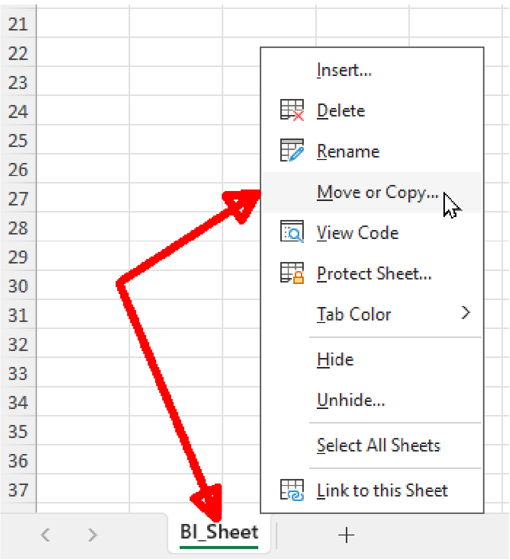
I am truly grateful for the support and guidance provided by the online community. I wish to list out some of them:

1. **Mark J. Nigrini** – Many of the forensic audit related (almost all!) functions are inspired from his AUTHORITATIVE book Forensic Analytics where he explained in detail about the algorithm and how to do them in MS Excel and MS Access. His research with Benford’s Law is mind blowing.
2. **Audit Monk** – A great resource of Excel/VBA script and algorithms for auditors.
3. **Mike Excelisfun Girvin** – Having 3500+ videos in YouTube, Mike Girvin had contributed a lot to Excel community and I can prodly say that he is my Guru in Power Query and Power Pivot.
4. **Owen Price** – His contributions in Lambda functions are available in his GitHub posts. I used one of his function as it is, with his permission. It is useful to calculate Levenshtein distance between two strings.
5. Many of my CA Professional Colleagues and students gave me ideas.

How to use this Template and its functions:

The template file 'BeIndian2.0.xltx' is the core library file containing an empty sheet called BI_Sheet. The Lambda Functions are stored in the workbook with a scope of the workbook itself. You can use this template and its functions in two ways:

1. **When you create a new file:** By double-clicking the template file, you can open a new file. You can then delete the empty sheet and add a new one for your calculations or simply rename the sheet to your preference. When saving the file, the system will prompt you to enter a new filename, and you can save the file with the '.xlsx' extension.
2. **When you want these functions in an existing workbook:** To add these functions to an existing workbook, copy the BI_Sheet into your existing file by right-clicking the sheet name and selecting 'Move or Copy' (see the image below). When you copy this sheet into your existing workbook, it will also copy all the functions. You can later delete the sheet while retaining the functions.



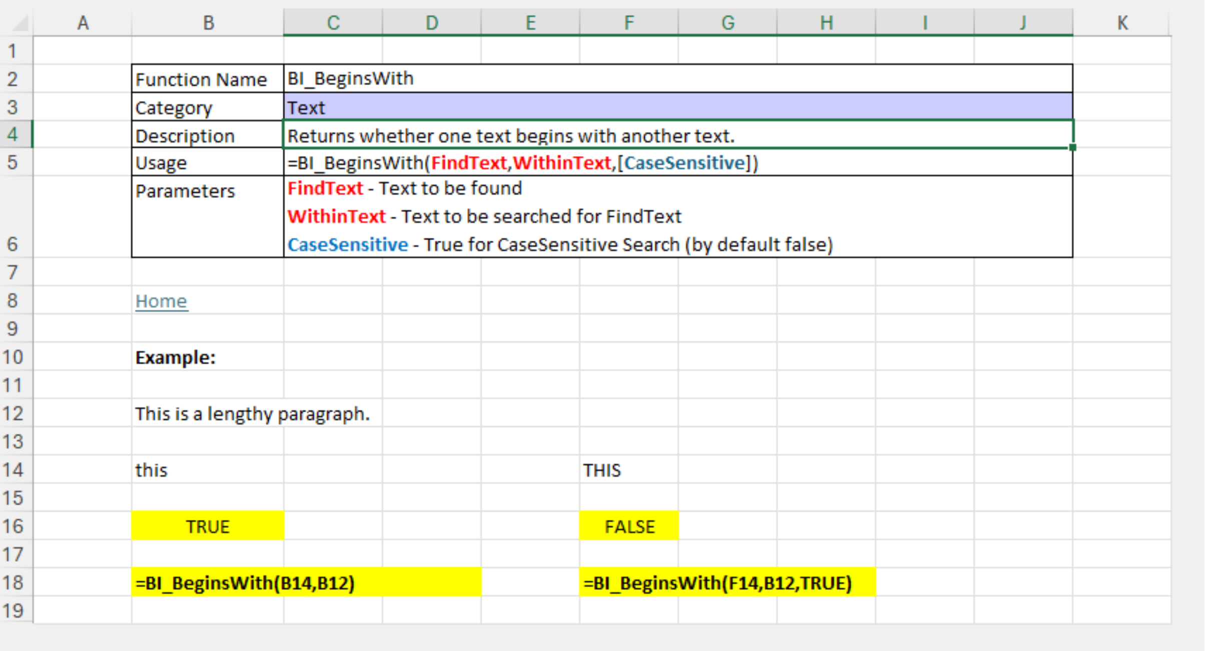
Demo and Help: All the functions in this template begin with the abbreviation "BI_" for easy identification. However, it is not necessary to always start typing "BI_". For example, if you want to calculate AltmanZScore, you can start typing "Altman," and Excel will list the "BI_AltmanZScore" function in the list. You can find detailed instructions on how these functions work in the attached Help file "BeIndian_Help.xlsx."

Syntax and color convention used: In every function, the required parameters are colored in red, and optional parameters are colored in blue and enclosed by square brackets []. For example, consider this function:

=BI_BeginsWith(**FindText**,**WithinText**,[**CaseSensitive**])

In the above example, "FindText" and "WithinText" parameters are required, while "CaseSensitive" searching is optional. It's worth noting that optional parameters have default values.

Each function is explained with suitable examples on the respective sheet. The formula text and the place where the formula should be entered are highlighted in yellow, as shown in the image below:



	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BeginsWith								
3		Category	Text								
4		Description	Returns whether one text begins with another text.								
5		Usage	=BI_BeginsWith(**FindText**,**WithinText**,[**CaseSensitive**])								
6		Parameters	**FindText** - Text to be found **WithinText** - Text to be searched for FindText **CaseSensitive** - True for CaseSensitive Search (by default false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		This is a lengthy paragraph.									
13											
14		this				THIS					
15											
16		TRUE				FALSE					
17											
18		**=BI_BeginsWith(B14,B12)**				**=BI_BeginsWith(F14,B12,TRUE)**					
19											

Navigation:

There is a link to the home sheet, and from the home sheet, there are links to every function sheet.

List of functions in **Be Indian – Excel Audit Tool Template**

Sl.No.	Function Name	Category
1	BI_AltmanZScore	Forensic Audit
2	BI_AltmanZScoreVariables	Forensic Audit
3	BI_AnovaSingleFactor	Statistics
4	BI_AnovaTwoFactorsWithoutReplication	Statistics
5	BI_AnsCombeQuartet	Statistics
6	BI_ArrayContains	Array
7	BI_ArrayCountif	Array
8	BI_ArrayFilter	Array
9	BI_ArrayInArray	Array
10	BI_ArrayMatch	Array
11	BI_ASort	Array
12	BI_AttributeUnitCount	Audit Sampling
13	BI_BeginsWith	Text
14	BI_BeneishMScore	Forensic Audit
15	BI_BeneishMScoreVariables	Forensic Audit
16	BI_BenfordLaw_FirstDigit	Forensic Audit
17	BI_BenfordLaw_FirstThreeDigits	Forensic Audit
18	BI_BenfordLaw_FirstTwoDigits	Forensic Audit
19	BI_BenfordLaw_LastTwoDigits	Forensic Audit
20	BI_BenfordLaw_SecondDigit	Forensic Audit
21	BI_BenfordLaw_SecondOrder	Forensic Audit

22	BI_BenfordLaw_SummaryTest	Forensic Audit
23	BI_BenfordLaw_ThirdDigit	Forensic Audit
24	BI_BottomN	Statistics
25	BI_BottomPercent	Statistics
26	BI_Consolidate	Data
27	BI_Contains	Text
28	BI_Correlation	Statistics
29	BI_Covariance	Statistics
30	BI_DateDif	Date
31	BI_Describe	Statistics
32	BI_DiscountedPayback	Financial
33	BI_EMISchedule	Financial
34	BI_EndsWith	Text
35	BI_ExactArray	Array
36	BI_ExponentialSmoothing	Statistics
37	BI_ExtractNumbers	Text
38	BI_ExtractText	Text
39	BI_FDRSchedule	Financial
40	BI_FillDown	Text
41	BI_FinancialYear	Date
42	BI_FinancialYearEnd	Date
43	BI_FinancialYearStart	Date
44	BI_FindDuplicates	Audit
45	BI_FindMissingNumbers	Audit
46	BI_FindOneOf	Audit
47	BI_FindUnique	Audit
48	BI_FormulaList	Audit
49	BI_Fuzzy	Audit
50	BI_GEL1Test	Forensic Audit
51	BI_GEL2Test	Forensic Audit
52	BI_GenerateRandom	Statistics

53	BI_GenerateRandomBetween	Statistics
54	BI_Histogram	Statistics
55	BI_IndianNum2Word	Indian
56	BI_Insert	Text
57	BI_IsCharAtoZ	Text
58	BI_IsLeap	Date
59	BI_IsValidPAN	Audit
60	BI_LuhnAlgorithm	Audit
61	BI_MonthlyCalendar	Date
62	BI_MovingAverage	Statistics
63	BI_MUS_Evaluate_OverStatement	Audit Sampling
64	BI_MUS_Evaluate_UnderStatement	Audit Sampling
65	BI_MUS_ExtractSample	Audit Sampling
66	BI_MUS_SampleSize	Audit Sampling
67	BI_Networkdays.Indian	Indian
68	BI_Num2Word	Text
69	BI_OhlsonsOScore	Forensic Audit
70	BI_OhlsonsOScoreVariables	Forensic Audit
71	BI_PaybackPeriod	Financial
72	BI_PiotroskiFScore	Forensic Audit
73	BI_PiotroskiFScoreVariables	Forensic Audit
74	BI_Pivot	Data
75	BI_Quarter	Date
76	BI_QuarterEnd	Date
77	BI_QuarterStart	Date
78	BI_Rank.Dense	Statistics
79	BI_Regression	Statistics
80	BI_RelativeSizeFactor	Forensic Audit
81	BI_ReverseText	Text
82	BI_RoundNumbersTest	Forensic Audit
83	BI_RowOccurence	Array

84	BI_RunningTotal	Data
85	BI_RupeeSymbol	Indian
86	BI_SameSameDifferentTest	Forensic Audit
87	BI_SameSameSameTest	Forensic Audit
88	BI_SLNSchedule	Financial
89	BI_SubsetNumberDuplicationTest	Forensic Audit
90	BI_Summary	Data
91	BI_TextSplitByPositions	Text
92	BI_TopN	Statistics
93	BI_TopPercent	Statistics
94	BI_Unpivot	Data
95	BI_UnpivotExceptFirst	Data
96	BI_WDVSchedule	Financial
97	BI_WordCount	Text
98	BI_Workday.Indian	Indian
99	BI_YearlyCalendar	Date
100	BI_ZScore	Statistics

About the tool:

Template Name:	**Be Indian Excel Audit Tool**
Version:	**2.0**
Author:	**CA S. Rathinagiri**
Copyright:	None. 100% Free without any warranties whatsoever
Date Published:	2023, October 25
Contact Information:	srathinagiri@gmail.com
Description:	Excel template containing 100+ Lambda functions useful for Audit Community.
Instructions:	You are ready to use this template when you 1. Open as a New File 2. Copy any sheet to existing workbook
Disclaimer:	This template comes absolutely with no warranties or guarantees of accuracy or fitness for a specific purpose.
Additional Information:	If you use this template please give your comments/suggestions to the above mail id

Download: You can download this template and help files from

1. My website.
2. My facebook page.
3. My LinkedIn page.
4. OneDrive Folder.

	A	B	C	D	E	F	G	H	I	J
1										
2		Function Name	**BI_AltmanZScore**							
3		Category	Forensic Audit							
4		Description	Calculates Altman Z Score. The Altman Z score determines the chances of bankruptcy for a company. The Altman Z score for companies with the highest bankruptcy risk is typically around -0.25. On the other hand, for companies with the lowest chance of bankruptcy, the Altman Z score value may rise to +4.48.							
5		Usage	=BI_AltmanZScore(**Data**)							
6		Parameters	**Data** - A column of Data as listed in BI_AltmanZScoreVariables()							
7										
8		Home								
9										
10		**Example:**								
11										
12		Working Capital		100000						
13		Total Assets		500000						
14		Retained Earnings		80000						
15		Earnings Before Interest and Taxes		130000						
16		Market Value		600000						
17		Total Liabilities		300000						
18		Sales		800000						
19										
20		Altman Z Score	4.122							
21		Verdict	>= 3 - Safe Zone - Low likelyhood of Bankruptcy							
22										
23		**=BI_AltmanZScore(D12:D18)**								
24										

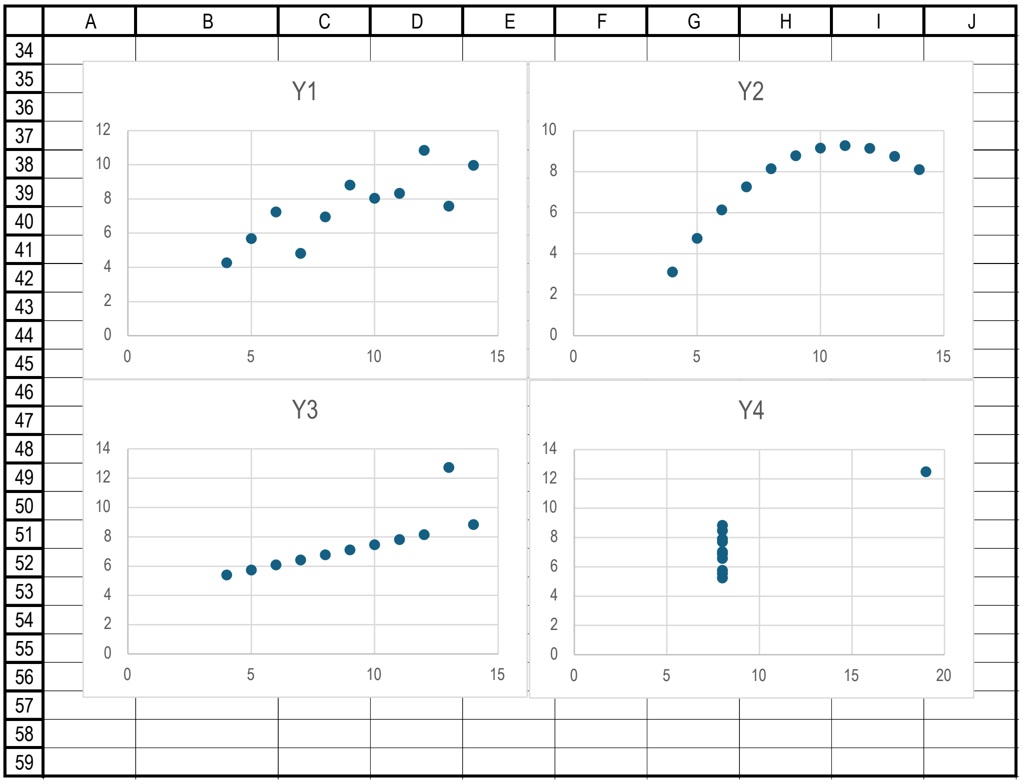
	A	B	C	D	E	F	G	H	I	J
1										
2		Function Name	BI_AltmanZScoreVariables							
3		Category	Forensic Audit							
4		Description	Lists out Altman Z Score Variables.							
5		Usage	=BI_AltmanZScoreVariables()							
6		Parameters	Nil							
7										
8		Home								
9										
10		**Example:**								
11										
12		Working Capital								
13		Total Assets								
14		Retained Earnings								
15		Earnings Before Interest and Taxes								
16		Market Value								
17		Total Liabilities								
18		Sales								
19										
20		**=BI_AltmanZScoreVariables()**								
21										

	A	B	C	D	E	F	G	H	I	J
1										
2		Function Name	BI_AnovaSingleFactor							
3		Category	Statistics							
4		Description	Calculates Anova (Analysis of Variance) Single Factor. This is one of the Data Analysis Toolpak features.							
5		Usage	=BI_AnovaSingleFactor(**TableData**, **Alpha**, **Labels**)							
6		Parameters	1. TableData 2. Alpha value (usually 0.05) 3. Labels (true if first row contains label)							
7										
8		Home								
9										
10										
11		Example:								
12			Revenue Through							
13		Period	Salesman	Advertisem	Website	Direct				
14		ஏப்ரல்-23	1092	1057	1089	1102				
15		மே-23	1139	1122	1163	1024				
16		ஜூன்-23	1040	1089	1030	1112				
17		ஜூலை-23	1173	1155	1079	1142				
18		ஆகஸ்ட்-23	1178	1034	1114	1031				
19		செப்டம்பர்-23	1182	1080	1172	1086				
20		அக்டோபர்23	1119	1122	1045	1169				
21		நவம்பர்-23	1031	1002	1140	1045				
22		டிசம்பர்23	1103	1036	1146	1118				
23										
24										
25		Anova: Single Factor								
26										
27		Summary								
28		Groups	Count	Sum	Average	Variance				
29		Salesman	9	10057	1117.444	3214.278				
30		Advertisements	9	9697	1077.444	2480.028				
31		Website	9	9978	1108.667	2609.5				
32		Direct	9	9829	1092.111	2524.361				
33										
34										
35		Anova								
36		Source of Variation	SS	DF	MS	F	P-value	F Crit		
37		Between Groups	8511.416667	3	2837.139	1.048059	0.384734	2.90112		
38		Within Groups	86625.33333	32	2707.042					
39										
40		Total	95136.75	35						
41										
42										
43		**=BI_AnovaSingleFactor(C13:F22,0.05,TRUE)**								
44										

	A	B	C	D	E	F	G	H	I
1									
2		Function Name	BI_AnovaTwoFactorsWithoutReplication						
3		Category	Statistics						
4		Description	Calculates Anova (Analysis of Variance) without replication. This is one of the Data Analysis Toolpak features.						
5		Usage	=BI_AnovaTwoFactorsWithoutReplication(**TableData**, **Alpha**, [**Labels**])						
6		Parameters	TableData - Data for Analysis Alpha - Alpha level (Usually 0.05) Labels - True if first row contains field labels. By default it is false						
7									
8		Home							
9									
10		Example:							
11									
12			Revenue Through						
13		Samples	Salesman	Advertisements	Website	Direct			
14		Men Customers	1190	1000	1079	1002			
15		Women Customers	1028	1044	1013	1114			
16		Men Customers	1098	1031	1168	1122			
17		Women Customers	1145	1014	1184	1171			
18		Men Customers	1125	1173	1103	1048			
19		Women Customers	1112	1178	1110	1069			
20		Men Customers	1152	1022	1143	1182			
21		Women Customers	1198	1151	1143	1038			
22		Men Customers	1128	1006	1199	1027			
23		Women Customers	1081	1000	1043	1115			
24		Men Customers	1130	1070	1190	1062			
25		Women Customers	1135	1075	1171	1092			
26		Men Customers	1093	1018	1045	1124			
27		Women Customers	1144	1142	1044	1032			
28		Men Customers	1139	1147	1083	1050			
29		Women Customers	1077	1154	1149	1091			
30									
31									

	A	B	C	D	E	F	G	H	I
32		Anova:Two-FactorWithoutReplication							
33									
34		SUMMARY	Count	Sum	Average	Variance			
35		Men Customers	4	4271	1067.75	7994.916667			
36		Women Customers	4	4199	1049.75	1994.916667			
37		Men Customers	4	4419	1104.75	3260.916667			
38		Women Customers	4	4514	1128.5	6089.666667			
39		Men Customers	4	4449	1112.25	2688.916667			
40		Women Customers	4	4469	1117.25	2032.916667			
41		Men Customers	4	4499	1124.75	4970.25			
42		Women Customers	4	4530	1132.5	4557.666667			
43		Men Customers	4	4360	1090	8116.666667			
44		Women Customers	4	4239	1059.75	2451.583333			
45		Men Customers	4	4452	1113	3556			
46		Women Customers	4	4473	1118.25	1874.25			
47		Men Customers	4	4280	1070	2258			
48		Women Customers	4	4362	1090.5	3699.666667			
49		Men Customers	4	4419	1104.75	2142.916667			
50		Women Customers	4	4471	1117.75	1555.583333			
51									
52		Salesman	16	17975	1123.4375	1793.729167			
53		Advertisements	16	17225	1076.5625	4705.0625			
54		Website	16	17867	1116.6875	3605.5625			
55		Direct	16	17339	1083.6875	2676.229167			
56									
57		Anova							
58		SourceofVariation	SS	df	MS	F	P-value	Fcrit	
59		Rows	40264.9375	15	2684.329167	0.797621313	0.673773	1.894875	
60		Columns	26290.6875	3	8763.5625	2.604004125	0.063442	2.811544	
61		Error	151443.8125	45	3365.418056				
62									
63		Total	217999.4375	63					
64									
65									
66		**=BI_AnovaTwoFactorsWithoutReplication(B13:F29,0.05,TRUE)**							
67									

	A	B	C	D	E	F	G	H	I	J
1										
2		Function Name	**BI_AnsCombeQuartet**							
3		Category	Statistics							
4		Description	Shows AnsCombe Quartet. Anscombe's quartet is a group of four data sets that are nearly identical in simple descriptive statistics, but there are peculiarities that fool the regression model once you plot each data set.							
5		Usage	=BI_AnsCombeQuartet()							
6		Parameters	None							
7										
8		Home								
9										
10		**Example:**								
11										
12		X1	Y1	X2	Y2	X3	Y3	X4	Y4	
13		10	8.04	10	9.14	10	7.46	8	6.58	
14		8	6.95	8	8.14	8	6.77	8	5.76	
15		13	7.58	13	8.74	13	12.74	8	7.71	
16		9	8.81	9	8.77	9	7.11	8	8.84	
17		11	8.33	11	9.26	11	7.81	8	8.47	
18		14	9.96	14	8.1	14	8.84	8	7.04	
19		6	7.24	6	6.13	6	6.08	8	5.25	
20		4	4.26	4	3.1	4	5.39	19	12.5	
21		12	10.84	12	9.13	12	8.15	8	5.56	
22		7	4.82	7	7.26	7	6.42	8	7.91	
23		5	5.68	5	4.74	5	5.73	8	6.89	
24		Descriptive Statistics								
25		Average X	9							
26		Average Y	7.5							
27		Sample Variance	11							
28		XY Correlation	0.816							
29		R Squared	0.67							
30		Linear Regression	3.00+0.50x							
31										
32		=BI_AnsCombeQuartet()								
33										


A
B
C
D
E
F
G
H
I
J
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
Y1
12
10
8
6
4
2
0
0
5
10
15
Y2
10
8
6
4
2
0
0
5
10
15
Y3
14
12
10
8
6
4
2
0
0
5
10
15
Y4
14
12
10
8
6
4
2
0
0
5
10
15
20

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ArrayContains								
3		Category	Array								
4		Description	Verifies whether one record of data contains in a table and returns true or								
5		Usage	=BI_ArrayContains(**Array1**, **Array2**)								
6		Parameters	Array1 - An Array of many rows and columns or table of data Array2 - A Row of data to verify								
7											
8		Home									
9											
10		**Example:**									
11											
12		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
13		1	A	4	79		4	D	80	25	
14		2	B	77	55						
15		3	C	89	44						
16		4	D	80	25						
17		5	E	51	62						
18		6	F	66	41						
19		7	G	41	33						
20											
21											
22		**TRUE**									
23											
24		**=BI_ArrayContains(B13:E19,G13:J13)**									
25											
26											
27		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
28		1	A	4	79		4	D	40	25	
29		2	B	77	55						
30		3	C	89	44						
31		4	D	80	25						
32		5	E	51	62						
33		6	F	66	41						
34		7	G	41	33						
35											
36											
37		**FALSE**									
38											
39		**=BI_ArrayContains(B28:E34,G28:J28)**									
40											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ArrayCountif								
3		Category	Array								
4		Description	Counts howmany times a record of data repeated in a table of data. If not found								
5		Usage	=BI_ArrayCountif(**Array**, **CriteriaArray**)								
6		Parameters	Array - Table of data or array Criteria Array - Array row to be counted for.								
7											
8		Home									
9											
10		**Example:**									
11											
12		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
13		1	A	4	79		5	E	51	62	
14		2	B	77	55						
15		3	C	89	44						
16		4	D	80	25						
17		5	E	51	62						
18		6	F	66	41						
19		5	E	51	62						
20											
21											
22		**2**									
23											
24		**=BI_ArrayCountif(B13:E19,G13:J13)**									
25											
26							RollNo	Name	Mark1	Mark2	
27							5	E	4	62	
28											
29		**#N/A**									
30											
31		**=BI_ArrayCountif(B13:E19,G27:J27)**									
32											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ArrayFilter								
3		Category	Array								
4		Description	Filters a Table for a multi-column criteria array by matching selected columns with the								
5		Usage	=BI_ArrayFilter(Array, CriteriaArray, [SelectedColumns])								
6		Parameters	Array - TableData/Array Criteria Array - Criteria Row SelectedColumns - Array of SelectedColumns (by default all the columns are selected)								
7											
8		Home									
9											
10		**Example:**									
11											
12	**1**	RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
13		1	A	4	79		5	E	51	62	
14		2	B	77	55						
15		3	C	89	44						
16		4	D	80	25						
17		5	E	51	41						
18		6	E	66	41						
19		5	E	51	62						
20											
21											
22		5	E	51	62						
23											
24											
25		**=BI_ArrayFilter(B13:E19,G13:J13)**									
26											
27	**2**	**When array not found**					RollNo	Name	Mark1	Mark2	
28							5	E	40	62	
29											
30		#N/A									
31											
32		**=BI_ArrayFilter(B13:E19,G28:J28)**									
33											
34	3	**Selected Columns**									
35							Name	Mark2			
36							E	41			
37		5	E	51	41						
38		6	E	66	41						
39											
40		**=BI_ArrayFilter(B13:E19,G36:H36,{2,4})**									
41											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ArrayInArray								
3		Category	Array								
4		Description	Filters only the available rows in an array found in another array. It is like SQL INNER JOIN								
5		Usage	=BI_ArrayInArray(**Array1**, **Array2**)								
6		Parameters	**Array1** - TableData or Array **Array2** - Array to be filtered								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13											
14		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
15		1	A	4	79		5	E	51	62	
16		2	B	77	55		**2**	B	77	55	
17		3	C	89	44		3	C	90	44	
18		4	D	80	25						
19		5	E	51	41						
20		6	E	66	41						
21		5	E	51	62						
22											
23											
24		2	B	77	55						
25		5	E	51	62						
26											
27		**=BI_ArrayInArray(B15:E21,G15:J17)**									
28											
29											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ArrayMatch								
3		Category	Array								
4		Description	Returns the row number of an array where a matching row is first found. Returns #NA								
5		Usage	=BI_ArrayMatch(**LookupValueArray**, **LookupArray**, [**SelectedColumns**])								
6		Parameters	**LookupValueArray** - Row to be lookedup **LookupArray** - TableData or Array **SelectedColumns** - An array of Selected Columns to be LookedUp (by default all the columns are selected)								
7											
8		Home									
9											
10		**Example:**									
11											
12	1	RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
13		1	A	4	79		5	E	51	62	
14		2	B	77	55						
15		3	C	89	44						
16		4	D	80	25						
17		5	E	51	41						
18		6	E	66	41						
19		5	E	51	62						
20											
21											
22		7									
23											
24		=BI_ArrayMatch(G13:J13,B13:E19)									
25											
26	2	When not found:					RollNo	Name	Mark1	Mark2	
27							5	E	81	62	
28		#N/A									
29											
30		=BI_ArrayMatch(G27:J27,B13:E19)									
31							Name	Mark1			
32	3	Only Selected Columns					C	89			
33											
34		3									
35											
36		=BI_ArrayMatch(G32:H32,B13:E19,{2,3})									
37											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ASort								
3		Category	Array								
4		Description	Returns the sorted array by all the columns in the order entered. Useful for sorting								
5		Usage	=BI_Asort(**Array**,[**Order**],[**ColumnIndex**])								
6		Parameters	**Array**: Multicolumn Array to be sorted **Order**: 1 for Ascending, -1 for Descending (by default Ascending) **ColumnIndex**: Upto which column to be sorted (by default all the columns)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Region	State	City							
13		Region1	State1	City1							
14		Region2	State2	City2							
15		Region3	State3	City3							
16		Region4	State1	City4							
17		Region1	State2	City5							
18		Region2	State3	City1							
19		Region3	State1	City2							
20		Region4	State2	City3							
21		Region1	State3	City4							
22		Region2	State1	City5							
23		Region3	State2	City1							
24		Region4	State3	City2							
25		Region1	State1	City3							
26		Region2	State2	City4							
27		Region3	State3	City5							
28		Region4	State1	City1							
29		Region1	State2	City2							
30		Region2	State3	City3							
31		Region3	State1	City4							
32		Region4	State2	City5							
33											
34											
35	1	**All columns Sorted by ascending**									
36		Region1	State1	City1		=BI_ASort(B13:D32)					
37		Region1	State1	City3							
38		Region1	State2	City2							
39		Region1	State2	City5							
40		Region1	State3	City4							
41		Region2	State1	City5							
42		Region2	State2	City2							
43		Region2	State2	City4							
44		Region2	State3	City1							
45		Region2	State3	City3							
46		Region3	State1	City2							
47		Region3	State1	City4							
48		Region3	State2	City1							
49		Region3	State3	City3							
50		Region3	State3	City5							
51		Region4	State1	City1							
52		Region4	State1	City4							
53		Region4	State2	City3							
54		Region4	State2	City5							
55		Region4	State3	City2							
56											

	A	B	C	D	E	F	G	H	I	J	K
57											
58	2	**All Columns sorted by descending**									
59		Region4	State3	City2		=BI_ASort(B13:D32,-1)					
60		Region4	State2	City5							
61		Region4	State2	City3							
62		Region4	State1	City4							
63		Region4	State1	City1							
64		Region3	State3	City5							
65		Region3	State3	City3							
66		Region3	State2	City1							
67		Region3	State1	City4							
68		Region3	State1	City2							
69		Region2	State3	City3							
70		Region2	State3	City1							
71		Region2	State2	City4							
72		Region2	State2	City2							
73		Region2	State1	City5							
74		Region1	State3	City4							
75		Region1	State2	City5							
76		Region1	State2	City2							
77		Region1	State1	City3							
78		Region1	State1	City1							
79											
80	3	Sort only upto second column									
81		Region1	State1	City1		=BI_ASort(B13:D32,,2)					
82		Region1	State1	City3							
83		Region1	State2	City5							
84		Region1	State2	City2							
85		Region1	State3	City4							
86		Region2	State1	City5							
87		Region2	State2	City2							
88		Region2	State2	City4							
89		Region2	State3	City1							
90		Region2	State3	City3							
91		Region3	State1	City2							
92		Region3	State1	City4							
93		Region3	State2	City1							
94		Region3	State3	City3							
95		Region3	State3	City5							
96		Region4	State1	City4							
97		Region4	State1	City1							
98		Region4	State2	City3							
99		Region4	State2	City5							
100		Region4	State3	City2							
101											

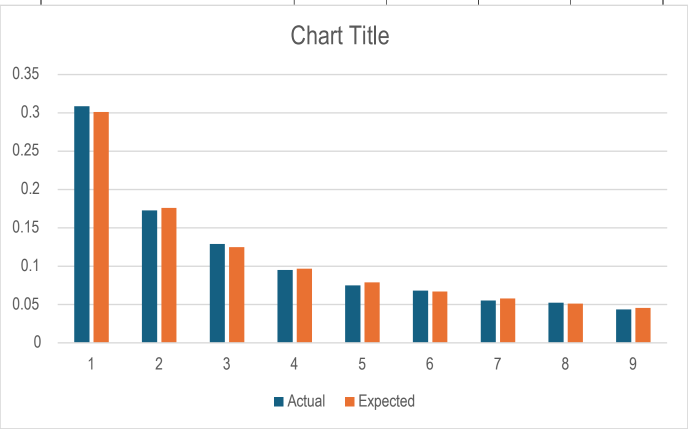
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_AttributeUnitCount								
3		Category	Audit Sampling								
4		Description	Returns Attribute Unit Sampling cumulative count of any attribute.								
5		Usage	=BI_AttributeUnitCount(**Data**)								
6		Parameters	**Data** - Any attribute column. It is better to keep an index column also to identify the row. See the example.								
7											
8		Home									
9											
10		**Example:**									
11											
12		1	Bookcases	1		**=BI_AttributeUnitCount(SalesTable[Sub-Category])**					
13		2	Chairs	1							
14		3	Labels	1							
15		4	Tables	1							
16		5	Storage	1							
17		6	Furnishings	1							
18		7	Art	1							
19		8	Phones	1							
20		9	Binders	1							
21		10	Appliances	1							
22		11	Tables	2							
23		12	Phones	2							
24		13	Paper	1							
25		14	Binders	2							
26		15	Appliances	2							
27		16	Binders	3							
28		17	Storage	2							
29		18	Storage	3							
30		19	Art	2							
31		20	Phones	3							
32		21	Binders	4							
33		22	Art	3							
34		23	Appliances	3							
35		24	Chairs	2							
36		25	Tables	3							
37		26	Binders	5							
38		27	Accessories	1							
39		28	Bookcases	2							
40		29	Binders	6							
41		30	Furnishings	2							
42		31	Envelopes	1							
43		32	Art	4							
44		33	Binders	7							
45		34	Art	5							
46		35	Paper	2							
47		36	Phones	4							
48		37	Furnishings	3							
49		38	Envelopes	2							
50		39	Bookcases	3							

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BeginsWith								
3		Category	Text								
4		Description	Returns whether one text begins with another text.								
5		Usage	=BI_BeginsWith(FindText,WithinText,[CaseSensitive])								
6		Parameters	FindText - Text to be found WithinText - Text to be searched for FindText CaseSensitive - True for CaseSensitive Search (by default false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		This is a lengthy paragraph.									
13											
14		this				THIS					
15											
16		TRUE				FALSE					
17											
18		**=BI_BeginsWith(B14,B12)**				**=BI_BeginsWith(F14,B12,TRUE)**					
19											

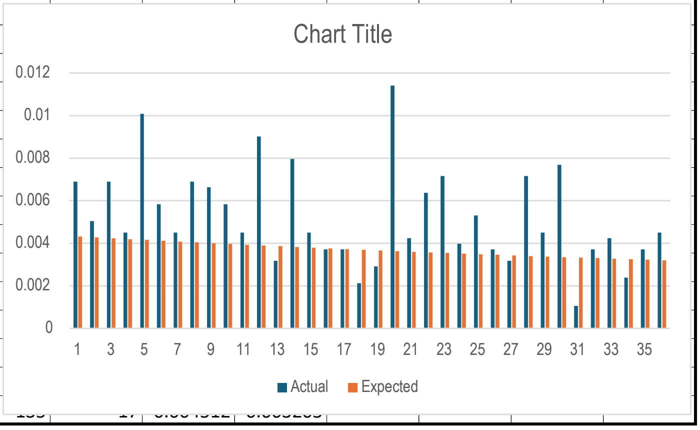
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BeneishMScore								
3		Category	Forensic Audit								
4		Description	Calculates Beneish M Score. Beneish's M-Score is a mathematical model that uses eight								
5		Usage	=BI_BeneishMScore(**LastYearData**,**CurrentYearData**)								
6		Parameters	**LastYearData**: LastYearData to be analyzed in the Beneish M Score Variables order **CurrentYearData**: CurrentYearData to be analyzed in the Beneish M Score Variables order								
7											
8		Home									
9											
10		**Example:**									
11											
12			2022	2023							
13		Net Sales	600000	680000							
14		Cost of Goods Sold	400000	420000							
15		Selling, General and Admin Expenses	60000	65000							
16		Net Income	100000	125000							
17		Property, Plant and Equipment	200000	250000							
18		Depreciation	100000	118000							
19		Current Assets	200000	240000							
20		Net Receivables	125000	130000							
21		Total Assets	900000	940000							
22		Long-tem Debt	400000	430000							
23		Current Liabilities	75000	80000							
24		Cash Flow from Operations	140000	160000							
25											
26		DSRI	0.91764706								
27		GMI	0.87179487								
28		AQI	0.86170213								
29		SGI	1.13333333								
30		DEPI	1.03954802								
31		SGAI	0.95588235								
32		LVGI	1.02799552								
33		TATA	-0.037234								
34		Beneish M Score 8 Variables	-2.7316324								
35		Verdict	Since M Score (8 Variables) is less than -2.22, the company is not likely to have manipulated its earnings								
36		Beneish M Score 5 Variables	-3.0851093								
37		Verdict	Since M Score (5 Variables) is less than -2.22, the company is not likely to have manipulated its earnings								
38											
39											
40		=BI_BeneishMScore(C13:C24,D13:D24)									
41											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BeneishMScoreVariables								
3		Category	Forensic Audit								
4		Description	Lists out Beneish M Score variables to make it easy to enter two years data for calculating Beneish M Score.								
5		Usage	=BI_BeneishMScoreVariables()								
6		Parameters	Nil								
7											
8		Home									
9											
10		**Example:**									
11											
12		Net Sales									
13		Cost of Goods Sold									
14		Selling, General and Admin Expenses									
15		Net Income									
16		Property, Plant and Equipment									
17		Depreciation									
18		Current Assets									
19		Net Receivables									
20		Total Assets									
21		Long-tem Debt									
22		Current Liabilities									
23		Cash Flow from Operations									
24											
25		=BI_BeneishMScoreVariables()									
26											

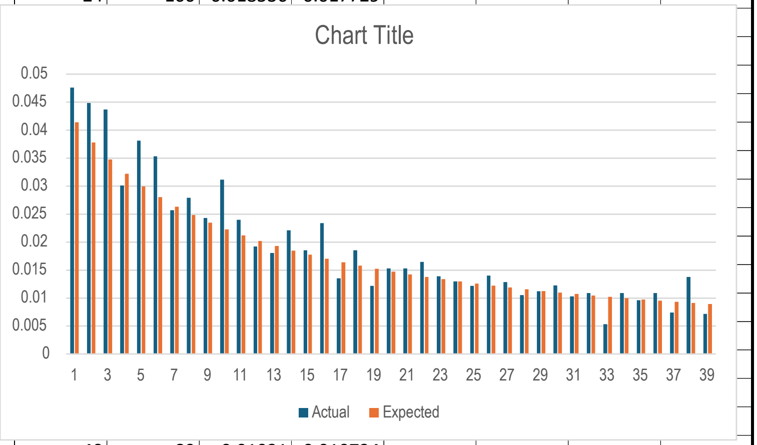
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_FirstDigit								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law First Digit - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_FirstDigit(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law First Digit								
7											
8		Home									
9											
10		**Example:**									
11											
12		261.96		Benford Law First Digit							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		1	3083	0.308732	0.30103				
15		957.5775		2	1727	0.172942	0.176091				
16		22.368		3	1287	0.12888	0.124939				
17		48.86		4	949	0.095033	0.09691				
18		7.28		5	748	0.074905	0.079181				
19		907.152		6	680	0.068095	0.066947				
20		18.504		7	552	0.055277	0.057992				
21		114.9		8	523	0.052373	0.051153				
22		1706.184		9	437	0.043761	0.045757				
23		911.424		Total	9986	1	1				
24		15.552		Mean Absolute Deviation			0.003114				
25		407.976									
26		68.81									
27		2.544		=BI_BenfordLaw_FirstDigit(B12#)							
28		665.88									
29		55.5									
30		8.56									
31		213.48									
32		22.72									
33		19.46									
34		60.34									
35		71.372									
36		1044.63									
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									



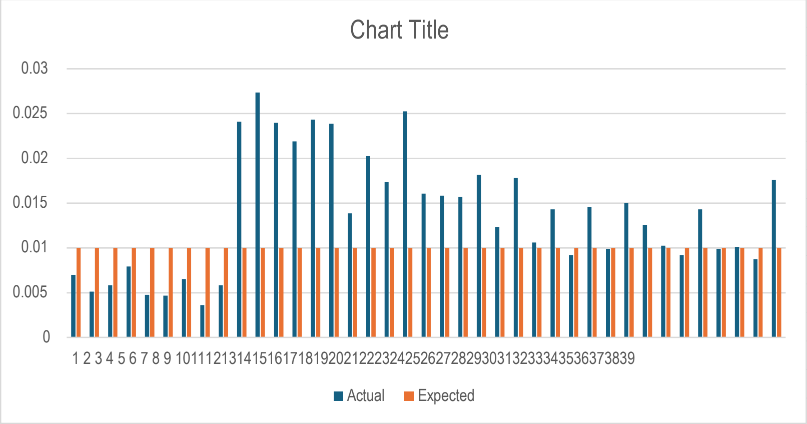
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_FirstThreeDigits								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law First Three Digits - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_FirstThreeDigits(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law First Three Digits								
7											
8		Home									
9											
10		**Example:**		=BI_BenfordLaw_FirstThreeDigits(B12#)							
11											
12		261.96		Benford Law First Three Digits							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		100	26	0.0069	0.004321				
15		957.5775		101	19	0.005042	0.004279				
16		22.368		102	26	0.0069	0.004237				
17		48.86		103	17	0.004512	0.004196				
18		7.28		104	38	0.010085	0.004156				
19		907.152		105	22	0.005839	0.004117				
20		18.504		106	17	0.004512	0.004078				
21		114.9		107	26	0.0069	0.00404				
22		1706.184		108	25	0.006635	0.004003				
23		911.424		109	22	0.005839	0.003966				
24		15.552		110	17	0.004512	0.00393				
25		407.976		111	34	0.009023	0.003895				
26		68.81		112	12	0.003185	0.00386				
27		2.544		113	30	0.007962	0.003826				
28		665.88		114	17	0.004512	0.003793				
29		55.5		115	14	0.003715	0.00376				
30		8.56		116	14	0.003715	0.003728				
31		213.48		117	8	0.002123	0.003696				
32		22.72		118	11	0.002919	0.003665				
33		19.46		119	43	0.011412	0.003634				
34		60.34		120	16	0.004246	0.003604				
35		71.372									
36		1044.63									
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									
49		113.328									



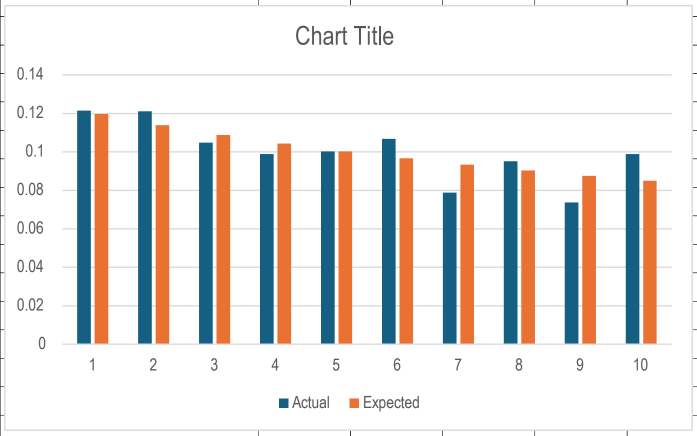
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_FirstTwoDigits								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law First Two Digits - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_FirstTwoDigits(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law First Two Digits								
7											
8		Home									
9											
10		**Example:**		**=BI_BenfordLaw_FirstTwoDigits(B12#)**							
11											
12		261.96		Benford Law First Two Digits							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		10	411	0.047614	0.041393				
15		957.5775		11	387	0.044833	0.037789				
16		22.368		12	377	0.043675	0.034762				
17		48.86		13	260	0.03012	0.032185				
18		7.28		14	329	0.038114	0.029963				
19		907.152		15	305	0.035334	0.028029				
20		18.504		16	222	0.025718	0.026329				
21		114.9		17	241	0.027919	0.024824				
22		1706.184		18	210	0.024328	0.023481				
23		911.424		19	269	0.031163	0.022276				
24		15.552		20	207	0.023981	0.021189				
25		407.976		21	166	0.019231	0.020203				
26		68.81		22	156	0.018072	0.019305				
27		2.544		23	191	0.022127	0.018483				
28		665.88		24	160	0.018536	0.017729				
29		55.5									
30		8.56									
31		213.48									
32		22.72									
33		19.46									
34		60.34									
35		71.372									
36		1044.63									
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858		40	89	0.01031	0.010724				
45		15.76		41	94	0.01089	0.010465				
46		29.472		42	46	0.005329	0.010219				
47		1097.544		43	94	0.01089	0.009984				
48		190.92		44	83	0.009615	0.00976				
49		113.328		45	94	0.01089	0.009545				
50		532.3992		46	64	0.007414	0.00934				
51		212.058		47	119	0.013786	0.009143				
52		371.168		48	62	0.007183	0.008955				



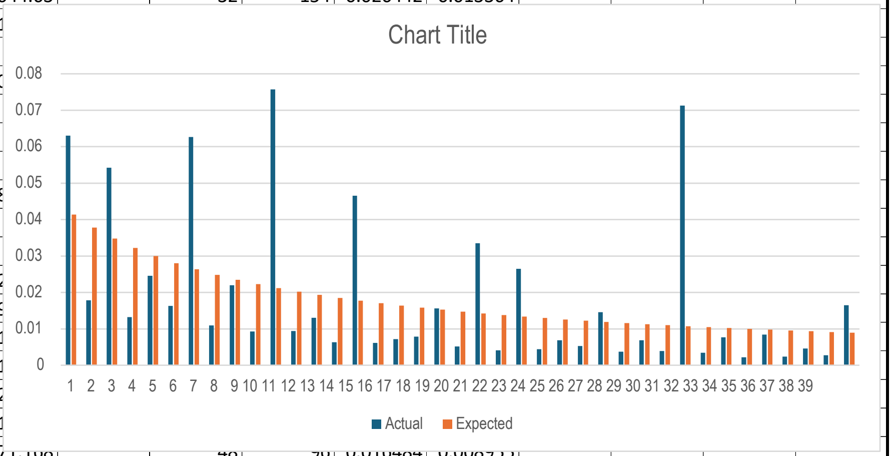
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_LastTwoDigits								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law Last Two Digits - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_LastTwoDigits(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law Last Two Digits								
7											
8		Home									
9				**=BI_BenfordLaw_LastTwoDigits(B12#)**							
10		**Example:**									
11											
12		261.96		Benford Law Last Two Digits							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		1	60	0.006984	0.01				
15		957.5775		2	44	0.005122	0.01				
16		22.368		3	50	0.00582	0.01				
17		48.86		4	68	0.007915	0.01				
18		7.28		5	41	0.004772	0.01				
19		907.152		6	40	0.004656	0.01				
20		18.504		7	56	0.006518	0.01				
21		114.9		8	31	0.003608	0.01				
22		1706.184		9	50	0.00582	0.01				
23		911.424		10	207	0.024095	0.01				
24		15.552		11	235	0.027354	0.01				
25		407.976		12	206	0.023979	0.01				
26		68.81		13	188	0.021883	0.01				
27		2.544		14	209	0.024328	0.01				
28		665.88		15	205	0.023862	0.01				
29		55.5		16	119	0.013852	0.01				
30		8.56		17	174	0.020254	0.01				
31		213.48		18	149	0.017344	0.01				
32		22.72		19	217	0.025259	0.01				
33		19.46		20	138	0.016063	0.01				
34		60.34		21	136	0.015831	0.01				
35		71.372		22	135	0.015714	0.01				
36		1044.63		23	156	0.018159	0.01				
37		11.648		24	106	0.012338	0.01				
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									
49		113.328									
50		532.3992									
51		212.058									
52		371.168									



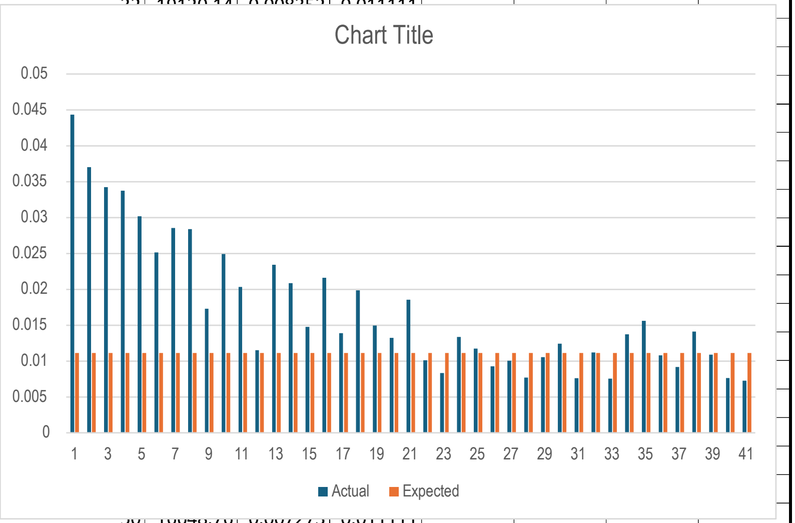
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_SecondDigit								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law Second Digit - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_SecondDigit(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law Second Digit								
7											
8		Home									
9											
10		**Example:**		=BI_BenfordLaw_SecondDigit(B12#)							
11											
12		261.96		Benford Law Second Digit							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		0	1049	0.121525	0.119679				
15		957.5775		1	1046	0.121177	0.11389				
16		22.368		2	905	0.104842	0.108821				
17		48.86		3	853	0.098818	0.10433				
18		7.28		4	866	0.100324	0.100308				
19		907.152		5	922	0.106812	0.096677				
20		18.504		6	681	0.078892	0.093375				
21		114.9		7	821	0.095111	0.090352				
22		1706.184		8	636	0.073679	0.08757				
23		911.424		9	853	0.098818	0.084997				
24		15.552		Total	9986	1	1				
25		407.976		Mean Absolute Deviation			0.007573				
26		68.81									
27		2.544									
28		665.88									
29		55.5									
30		8.56									
31		213.48									
32		22.72									
33		19.46									
34		60.34									
35		71.372									
36		1044.63									
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									
49		113.328									
50		532.3992									
51		212.058									
52		371.168									
53		147.168									
54		77.88									
55		95.616									
56		45.98									



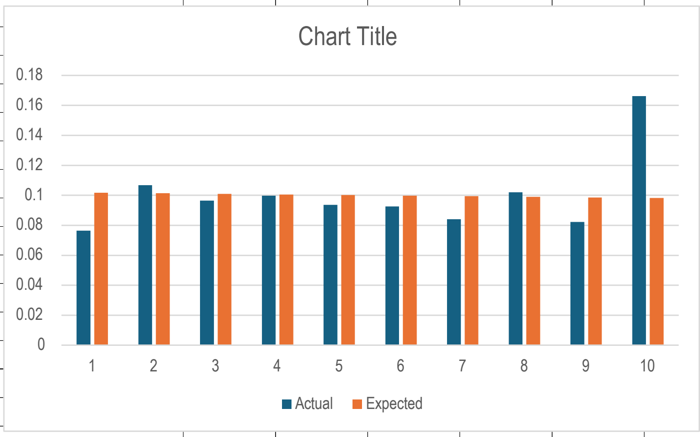
	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_SecondOrder								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law Second Order - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_SecondOrder(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law Second Order								
7											
8		Home									
9											
10		**Example:**		**=BI_BenfordLaw_SecondOrder(B12#)**							
11											
12		261.96		Benford Law Second Order							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		10	367	0.063015	0.041393				
15		957.5775		11	104	0.017857	0.037789				
16		22.368		12	316	0.054258	0.034762				
17		48.86		13	77	0.013221	0.032185				
18		7.28		14	143	0.024554	0.029963				
19		907.152		15	95	0.016312	0.028029				
20		18.504		16	365	0.062672	0.026329				
21		114.9		17	64	0.010989	0.024824				
22		1706.184		18	128	0.021978	0.023481				
23		911.424		19	54	0.009272	0.022276				
24		15.552		20	441	0.075721	0.021189				
25		407.976		21	55	0.009444	0.020203				
26		68.81		22	76	0.013049	0.019305				
27		2.544		23	37	0.006353	0.018483				
28		665.88		24	271	0.046532	0.017729				
29		55.5		25	36	0.006181	0.017033				
30		8.56		26	42	0.007212	0.01639				
31		213.48		27	46	0.007898	0.015794				
32		22.72		28	91	0.015625	0.01524				
33		19.46		29	30	0.005151	0.014723				
34		60.34		30	195	0.033482	0.01424				
35		71.372		31	24	0.004121	0.013788				
36		1044.63		32	154	0.026442	0.013364				
37		1									
38											
39		30									
40											
41											
42											
43		8									
44											
45											
46		2									
47		109									
48		1									
49		11									
50		532									
51		21									
52		371.168		48	96	0.016484	0.008955				



	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_SummaryTest								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law SummaryTest - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_SummaryTest(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law Summary Test								
7											
8		Home									
9											
10		**Example:**		**=BI_BenfordLaw_SummaryTest(B12#)**							
11											
12		261.96		Benford Law Summary Test							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		10	101461.4	0.044323	0.011111				
15		957.5775		11	84729.58	0.037013	0.011111				
16		22.368		12	78345.78	0.034225	0.011111				
17		48.86		13	77263.85	0.033752	0.011111				
18		7.28		14	69118.62	0.030194	0.011111				
19		907.152		15	57582.59	0.025155	0.011111				
20		18.504		16	65372.37	0.028557	0.011111				
21		114.9		17	65028.51	0.028407	0.011111				
22		1706.184		18	39610.71	0.017304	0.011111				
23		911.424		19	57078.27	0.024934	0.011111				
24		15.552		20	46527.8	0.020325	0.011111				
25		407.976		21	26372.65	0.011521	0.011111				
26		68.81		22	53660.39	0.023441	0.011111				
27		2.544		23	47768.79	0.020867	0.011111				
28		665.88		24	33795.34	0.014763	0.011111				
29		55.5		25	49513.29	0.021629	0.011111				
30		8.56		26	31789.09	0.013887	0.011111				
31		213.48		27	45476.56	0.019866	0.011111				
32		22.72		28	34288.38	0.014979	0.011111				
33		19.46		29	30312.02	0.013242	0.011111				
34		60.34		30	42450.17	0.018544	0.011111				
35		71.372		31	23198.5	0.010134	0.011111				
36		1044.63		32	19130.14	0.008352	0.011111				
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									
49		113.328									
50		532.3992									
51		212.058									
52		371.168									
53		147.168									
54		77.88		50	10048.76	0.007275	0.011111				



	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BenfordLaw_ThirdDigit								
3		Category	Forensic Audit								
4		Description	Calculates Benford Law Third Digit - Frequency, Actual and Expected values along with Mean Abstract Deviation.								
5		Usage	=BI_BenfordLaw_ThirdDigit(**BenfordRange**)								
6		Parameters	**BenfordRange** - A Column of Numbers to analyze for Benford Law Third Digit								
7											
8		Home									
9											
10		**Example:**		**=BI_BenfordLaw_ThirdDigit(B12#)**							
11											
12		261.96		Benford Law Third Digit							
13		731.94		Digit	Frequency	Actual	Expected				
14		14.62		0	391	0.076367	0.101784				
15		957.5775		1	547	0.106836	0.101376				
16		22.368		2	494	0.096484	0.100972				
17		48.86		3	511	0.099805	0.100573				
18		7.28		4	479	0.093555	0.100178				
19		907.152		5	474	0.092578	0.099788				
20		18.504		6	430	0.083984	0.099401				
21		114.9		7	522	0.101953	0.099019				
22		1706.184		8	421	0.082227	0.098641				
23		911.424		9	851	0.166211	0.098267				
24		15.552		Total	9986	1	1				
25		407.976		Mean Absolute Deviation			0.015268				
26		68.81									
27		2.544									
28		665.88									
29		55.5									
30		8.56									
31		213.48									
32		22.72									
33		19.46									
34		60.34									
35		71.372									
36		1044.63									
37		11.648									
38		90.57									
39		3083.43									
40		9.618									
41		124.2									
42		3.264									
43		86.304									
44		6.858									
45		15.76									
46		29.472									
47		1097.544									
48		190.92									
49		113.328									
50		532.3992									
51		212.058									
52		371.168									
53		147.168									
54		77.88									



	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BottomN								
3		Category	Statistics								
4		Description	Returns Bottom N ranked rows of a column of data.								
5		Usage	=BI_BottomN(**DataRange**, **IndexColumn**, **N**)								
6		Parameters	**DataRange** - TableData or Array **IndexColumn** - Column Number to be classified **N** - Number of Bottom Records								
7											
8		Home									
9											
10		**Example:**									
11											
12		5									
13											
14		4102	US-2017-10ZC-281910		0.444						
15		9293	CA-2017-12RS1-19765		0.556						
16		8659	CA-2016-16KB-116600		0.836						
17		4712	CA-2014-11JO-15280		0.852						
18		2107	US-2014-15EG-34965		0.876						
19											
20											
21											
22											
23											
24											
25		**=BI_BottomN(CHOOSECOLS(SalesTable,1,2,6,12),4,B12)**									
26											
27											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_BottomPercent								
3		Category	Statistics								
4		Description	Returns Bottom N Percent of a column of Data.								
5		Usage	=BI_BottomPercent(**Data**, **IndexColumn**, **Percent**)								
6		Parameters	**Data** - TableData or Array **IndexColumn** - Column Number to be classified **Percent** - Percentage of data to be extracted								
7											
8		Home									
9											
10		**Example:**									
11											
12		1%		**=BI_BottomPercent(CHOOSECOLS(SalesTable,1,2,6,12),4,B12)**							
13											
14		4102	US-2017-102288ZC-21910		0.444						
15		9293	CA-2017-124AS-119765		0.556						
16		8659	CA-2016-168KB-616600		0.836						
17		4712	CA-2014-112JO-015280		0.852						
18		2107	US-2014-152EG-314965		0.876						
19		7549	CA-2014-103AM-912715		0.898						
20		8034	CA-2015-119MV-017485		0.984						
21		2762	CA-2017-126NK-318490		0.99						
22		8025	CA-2014-129JM-814860		1.044						
23		1333	CA-2014-120MN-717935		1.08						
24		977	US-2017-100TD-220995		1.08						
25		4934	CA-2015-106ZD-721910		1.08						
26		988	CA-2015-146TS-821340		1.112						
27		2606	CA-2014-112KM-116450		1.167						
28		4875	CA-2017-164KD-416645		1.188						
29		1686	CA-2017-149DK-812835		1.188						
30		6090	US-2017-162EG-618745		1.188						
31		1113	US-2016-110DH-513945		1.192						
32		3326	CA-2014-166SB-016270		1.234						
33		5207	CA-2015-102TM-521010		1.24						
34		76	US-2017-118KD-316600		1.248						
35		8966	CA-2017-106CC-912370		1.248						
36		2347	CA-2016-159JT-317110		1.272						
37		2428	CA-2016-169DJ-217515		1.344						
38		3767	CA-2016-163BM-512955		1.344						
39		7118	CA-2014-143GH-615085		1.344						
40		9383	CA-2014-154AC-512670		1.344						
41		5857	CA-2015-112AH-110690		1.362						
42		6598	CA-2014-169SV-220785		1.365						
43		6990	CA-2017-165DK-913375		1.392						
44		9225	CA-2017-121EM-614290		1.408						
45		7357	CA-2016-114MZ-817335		1.408						
46		2214	CA-2017-128JS-718505		1.44						
47		1066	CA-2015-161TS-221160		1.448						
48		4782	CA-2014-159SG-210725		1.476						
49		9305	US-2015-168MP-317965		1.476						
50		3214	CA-2016-131JW-021235		1.504						

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Consolidate								
3		Category	Data								
4		Description	Returns consolidation of Multiple Ranges like Data->Consolidate feature. Upto 10 ranges can be consolidated.								
5		Usage	=BI_Consolidate(**Function**, **DataRange1**, [**DataRange2**],…,[**DataRange10**])								
6		Parameters	Function: Summary Function to be applied like Subtotal() DataRange1,DataRange2,…,**DataRange10**: Upto 10 DataRanges can be consolidated by Row and Column Labels								
7											
8		Home									
9											
10		**Example:**									
11											
12		East	2020	2021	2022			2020	2021	2022	
13		Purchase	15000	16000	18000		Building Maintenance	3300	4400	5500	
14		Sales	18000	20000	22000		General Expenses	800	900	1000	
15		Rent	3000	4000	5000		Postage	1980	2640	3300	
16		Salary	6000	8000	9000		Purchase	51300	54720	61560	
17		Stationery	1500	2000	2500		Rent	6960	9280	11600	
18							Salary	20520	27360	30780	
19							Sales	61560	68400	75240	
20		West	2020	2021	2022		Stationery	3150	4200	5250	
21		Purchase	16500	17600	19800						
22		Sales	19800	22000	24200						
23		Building Maintenance	3300	4400	5500		**=BI_Consolidate(9,B12:E17,B20:E25,B28:E34)**				
24		Salary	6600	8800	9900						
25		Stationery	1650	2200	2750						
26											
27											
28		South	2020	2021	2022						
29		General Expenses	800	900	1000						
30		Sales	23760	26400	29040						
31		Rent	3960	5280	6600						
32		Salary	7920	10560	11880						
33		Postage	1980	2640	3300						
34		Purchase	19800	21120	23760						
35											
36											
37											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Contains								
3		Category	Text								
4		Description	Returns whether one text contains another text.								
5		Usage	=BI_Contains(**FindText**,**WithinText**,[**CaseSensitive**])								
6		Parameters	FindText - Text to be found WithinText - Text to be searched for FindText CaseSensitive - True for CaseSensitive (By default the value is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		This is a lengthy paragraph.									
13											
14		lengthy				Lengthy					
15											
16		TRUE				FALSE					
17											
18		**=BI_Contains(B14,B12)**				**=BI_Contains(F14,B12,TRUE)**					
19											
20											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Correlation								
3		Category	Statistics								
4		Description	Returns a Correlation Table of many attributes. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_Correlation(**Data**,[**Labels**])								
6		Parameters	**Data** - Data for Correlation between variables in Columns **Labels** - True if the first row contains row labels. (by default the value is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13			Revenue Through								
14		Period	Salesman	Advertisements	Website	Direct					
15		ஏப்ரல்-23	1052	1193	1000	1132					
16		மே-23	1005	1064	1050	1116					
17		ஜூன்-23	1069	1001	1178	1145					
18		ஜூலை-23	1084	1142	1123	1153					
19		ஆக்ஸ்ட-23	1148	1055	1052	1161					
20		செப்டம்பர்-23	1082	1126	1173	1135					
21		அக்டோபர்-23	1099	1046	1151	1088					
22		நவம்பர்-23	1035	1079	1015	1183					
23		டிம்பர்23	1169	1081	1171	1066					
24											
25											
26		Correlation	Salesman	Advertisements	Website	Direct					
27		Salesman	1								
28		Advertisements	-0.13169	1							
29		Website	0.46049	-0.372	1						
30		Direct	-0.38908	0.082919	-0.48234	1					
31											
32		Correlation	Attributes	Values							
33		Website	Salesman	0.46049							
34		Direct	Advertisements	0.082919							
35		Advertisements	Salesman	-0.13169							
36		Website	Advertisements	-0.372							
37		Direct	Salesman	-0.38908							
38		Direct	Website	-0.48234							
39											
40											
41		**=BI_Correlation(C14:F23,TRUE)**									
42											
43											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Covariance								
3		Category	Statistics								
4		Description	Returns a Covariance Table of many attributes. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_Covariance(**Data**,[**Labels**])								
6		Parameters	**Data** - Data for finding Covariance between variables in Columns **Labels** - True if the first row contains row labels. (by default the value is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12			Revenue Through								
13		Period	Salesman	Advertisements	Website	Direct					
14		ஏ்பர்ல-23	1068	1001	1137	1150					
15		மே-23	1104	1013	1156	1122					
16		ஜூன-23	1178	1038	1168	1034					
17		ஜூலை-23	1084	1179	1117	1025					
18		ஆக்ஸ்ட-23	1163	1159	1081	1178					
19		செப்டம்ப்ர-23	1156	1046	1145	1156					
20		அ்கமடோப்ர-23	1076	1063	1027	1030					
21		நவ்மப்ர-23	1001	1019	1168	1103					
22		டெிம்பர23	1020	1172	1005	1185					
23											
24											
25		Covariance	Salesman	Advertisements	Website	Direct					
26		Salesman	3442.691								
27		Advertisements	-67.4074	4674							
28		Website	954.7531	-2464.37	3286.691						
29		Direct	-262.877	364.963	-595.901	3728.173					
30											
31											
32		**=BI_Covariance(C13:F22,TRUE)**									
33											
34											
35											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_DateDif								
3		Category	Date								
4		Description	Returns the various combinations of Date Differences from starting date to ending date. All the DateDif() function parameters are demonstrated.								
5		Usage	=BI_DateDif(**StartDate**,**EndDate**)								
6		Parameters	**StartDate** - Starting Date of a Period **EndDate** - Ending Date of a Period								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13		Starting Date	01-01-2000								
14		Ending Date	24-10-2023								
15											
16		Years	Y	23							
17		Months	YM	9							
18		Days	MD	23							
19		Total Months	M	285							
20		Total Days	D	8697							
21		Days after year	YD	296							
22											
23											
24		**=BI_DateDif(C13,C14)**									
25											
26											
27											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Describe								
3		Category	Statistics								
4		Description	Returns the descriptive statistics of Data.								
5		Usage	=BI_Describe(**Range**)								
6		Parameters	Range - A Column of Data								
7											
8		Home									
9											
10		**Example:**									
11											
12		Descriptive Summary Statistics									
13		Average	229.858								
14		Median	54.49								
15		Mode	12.96								
16		St.Dev. P.	623.2139								
17		St. Dev. S	623.2451								
18		Std. Error	6.234322								
19		Population Variance	388395.6								
20		Sample Variance	388434.5								
21		Kurtosis	305.3118								
22		Skewness.P	12.97081								
23		Skewness S	12.97275								
24		Range	22638.04								
25		Maximum	22638.48								
26		Minimum	0.444								
27		Sum	2297201								
28		Count	9994								
29		Quartile1	17.28								
30		Quartile3	209.94								
31		IQR	192.66								
32		1.5 Times IQR	288.99								
33		Whisker Lower Bound	0.444								
34		Whisker Outer Bound	498.93								
35											
36											
37		**=BI_Describe(SalesTable[Sales])**									
38											
39											
40											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_DiscountedPayback								
3		Category	Financial								
4		Description	Returns the discounted payback period using a discount rate.								
5		Usage	=BI_DiscountedPayback(**CashFlows**,**DiscountRate**)								
6		Parameters	**CashFlows** - A Range of Cashflows (negative - outflow, positive - inflow)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Year	CashFlow								
13		0	-1000000								
14		1	200000								
15		2	200000								
16		3	200000								
17		4	200000								
18		5	250000								
19		6	250000								
20		7	250000								
21		8	250000								
22											
23		Discount Rate	9%								
24											
25		6.296188743									
26											
27		=BI_DiscountedPayback(C13:C21,C23)									
28											
29											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_EMISchedule								
3		Category	Financial								
4		Description	Returns EMI Schedule of a loan								
5		Usage	=BI_EMISchedule(**Principal**,**Period**,**InterestRate**)								
6		Parameters	**Principal** - Loan Principal Amount **Period** - Constant number of payments **InterestRate** - Interest Rate per compounding/Instalment period								
7											
8		Home									
9											
10		**Example:**									
11											
12		Loan Amount	100000								
13		Number of Instalments	24	months							
14		Rate of Interest per month	1%	per month							
15											
16		EMI Amount	4707.347								
17		Period	Op.Balance	Interest	Total	Instalment	Cl.Balance				
18		1	100000	1000	101000	4707.347	96292.65				
19		2	96292.65	962.9265	97255.58	4707.347	92548.23				
20		3	92548.23	925.4823	93473.71	4707.347	88766.37				
21		4	88766.37	887.6637	89654.03	4707.347	84946.68				
22		5	84946.68	849.4668	85796.15	4707.347	81088.8				
23		6	81088.8	810.888	81899.69	4707.347	77192.34				
24		7	77192.34	771.9234	77964.27	4707.347	73256.92				
25		8	73256.92	732.5692	73989.49	4707.347	69282.14				
26		9	69282.14	692.8214	69974.96	4707.347	65267.62				
27		10	65267.62	652.6762	65920.29	4707.347	61212.95				
28		11	61212.95	612.1295	61825.07	4707.347	57117.73				
29		12	57117.73	571.1773	57688.9	4707.347	52981.56				
30		13	52981.56	529.8156	53511.37	4707.347	48804.03				
31		14	48804.03	488.0403	49292.07	4707.347	44584.72				
32		15	44584.72	445.8472	45030.57	4707.347	40323.22				
33		16	40323.22	403.2322	40726.45	4707.347	36019.1				
34		17	36019.1	360.191	36379.3	4707.347	31671.95				
35		18	31671.95	316.7195	31988.67	4707.347	27281.32				
36		19	27281.32	272.8132	27554.13	4707.347	22846.79				
37		20	22846.79	228.4679	23075.25	4707.347	18367.91				
38		21	18367.91	183.6791	18551.59	4707.347	13844.24				
39		22	13844.24	138.4424	13982.68	4707.347	9275.334				
40		23	9275.334	92.75334	9368.087	4707.347	4660.74				
41		24	4660.74	46.6074	4707.347	4707.347	4.55E-12				
42		Total		12976.33		112976.3					
43											
44		=BI_EMISchedule(C12,C13,C14)									
45											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_EndsWith								
3		Category	Text								
4		Description	Returns whether one text ends with another text.								
5		Usage	=BI_EndsWith(**FindText**,**WithinText**,[**CaseSensitive**])								
6		Parameters	**FindText** - Text to be found **WithinText** - Text to be searched for FindText **CaseSensitive** - True for Case Sensitive Search (by default it is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		This is a lengthy paragraph									
13											
14		paragraph			Paragraph						
15											
16		TRUE			FALSE						
17											
18		=BI_EndsWith(B14,B12)			=BI_EndsWith(E14,B12,TRUE)						
19											
20											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ExactArray								
3		Category	Array								
4		Description	Compares two arrays and returns whether both the arrays are equal.								
5		Usage	=BI_ExactArray(**Array1**,**Array2**,[**CaseSensitive**])								
6		Parameters	**Array1** - First Array to be verified. **Array2** - Second Array to be verified. **CaseSensitive** - True for CaseSensitive verification (by default it is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
13		1	A	4	79		1	a	4	79	
14		2	B	77	55		2	B	77	55	
15		3	C	89	44		3	C	89	44	
16		4	D	80	25		4	D	80	25	
17		5	E	51	41		5	E	51	41	
18		6	E	66	41		6	E	66	41	
19		5	E	51	62		5	E	51	62	
20											
21											
22		TRUE					FALSE				
23											
24		=BI_ExactArray(B13:E19,G13:J19)					=BI_ExactArray(B13:E19,G13:J19,TRUE)				
25											
26											
27		RollNo	Name	Mark1	Mark2		RollNo	Name	Mark1	Mark2	
28		1	A	4	79		1	A	4	79	
29		2	B	77	55		2	B	77	55	
30		3	C	89	44		3	C	89	44	
31		4	D	80	25		4	D	40	25	
32		5	E	51	41		5	E	51	41	
33		6	E	66	41		6	E	66	41	
34		5	E	51	62		5	E	51	62	
35											
36											
37		FALSE									
38											
39		=BI_ExactArray(B28:E34,G28:J34)									
40											

Function Name	BI_ExponentialSmoothing
Category	Statistics
Description	Returns Exponential Smoothened data after applying Damping Factor. This is one of the Data Analysis
Usage	=BI_ExponentialSmoothing(**Data**,[**DampingFactor**],[**Labels**])
Parameters	**Data** - A column of timeline data **DampingFactor** - Smoothening factor (by default it is 0.3) **Labels** - True for data with first row as label. (by default it is false)

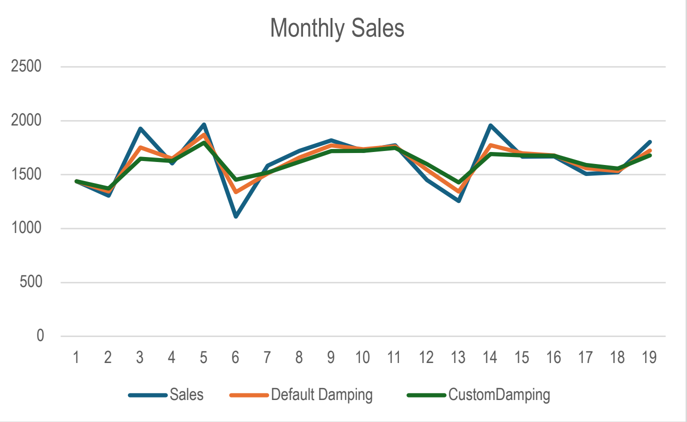
Home

Example:

			0.50
Period	Sales	Default Damping	CustomDamping
ஏ்பர்ல-22	1438	1438.00	1438.00
மே-22	1306	1345.60	1372.00
ஜூ்ன-22	1927	1752.58	1649.50
ஜூலை-22	1605	1649.27	1627.25
ஆக்ஸ்ட-22	1965	1870.28	1796.13
செ்பட்மப்ர-22	1111	1338.78	1453.56
அ்கமடோப்ர22	1584	1510.44	1518.78
நவ்மப்ர-22	1722	1658.53	1620.39
டி்மப்ர22	1819	1770.86	1719.70
ஜனவிர-23	1722	1736.66	1720.85
பிபரவிர-23	1774	1762.80	1747.42
மே◌ர்ச23	1452	1545.24	1599.71
ஏ்பர்ல-23	1256	1342.77	1427.86
மே-23	1957	1772.73	1692.43
ஜூ்ன-23	1667	1698.72	1679.71
ஜூலை-23	1670	1678.62	1674.86
ஆக்ஸ்ட-23	1507	1558.48	1590.93
செ்பட்மப்ர-23	1524	1534.35	1557.46
அ்கமடோப்ர23	1804	1723.10	1680.73

=BI_ExponentialSmoothing(C13:C32,,TRUE)

=BI_ExponentialSmoothing(C13:C32,E12,TRUE)



	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ExtractNumbers								
3		Category	Text								
4		Description	Extracts all the numbers from a text string.								
5		Usage	=BI_ExtractNumbers(**Text**)								
6		Parameters	Text - Text from which numbers to be extracted								
7											
8		Home									
9											
10		**Example:**									
11											
12		Rent paid by cheque number 125477 to Arumugam									
13											
14		125477									
15											
16		**=BI_ExtractNumbers(B12)**									
17											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ExtractText								
3		Category	Text								
4		Description	Extracts all the characters from a text string.								
5		Usage	=BI_ExtractText(**Text**)								
6		Parameters	Text - Text from which text to be extracted								
7											
8		Home									
9											
10		**Example:**									
11											
12		XYZ 14352 ABC									
13											
14		XYZ ABC									
15											
16		**=BI_ExtractText(B12)**									
17											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FDRSchedule								
3		Category	Financial								
4		Description	Returns the Fixed Deposit Schedule of Interest calculations and maturiy value. It is assumed that interest is paid on quarterly compounding (as in Indian Banking systems)								
5		Usage	=BI_FDRSchedule(**DepositAmount**,**InterestRate**,**Stdate**,**Enddate**)								
6		Parameters	**DepositAmount** - Present Value or initial deposit **InterestRate** - Interest Rate per annum **StDate** - Starting Date of the Fixed Deposit **EndingDate** - Maturity Date of the Fixed Deposit								
7											
8		Home									
9											
10		**Example:**									
11											
12		Investment	1000000								
13		Interest Rate	6%	p.a.							
14		Starting Date	15-10-2023		**=BI_FDRSchedule(C12,C13,C14,C15)**						
15		Maturity Date	01-02-2026								
16											
17		Date	Days	Opening	Interest	Closing					
18		15-10-2023	77	1000000	12657.53	1012658					
19		31-12-2023	91	1012658	15148.25	1027806					
20		31-03-2024	91	1027806	15374.85	1043181					
21		30-06-2024	92	1043181	15776.32	1058957					
22		30-09-2024	92	1058957	16014.91	1074972					
23		31-12-2024	90	1074972	15903.69	1090876					
24		31-03-2025	91	1090876	16318.3	1107194					
25		30-06-2025	92	1107194	16744.41	1123938					
26		30-09-2025	92	1123938	16997.64	1140936					
27		31-12-2025	32	1140936	6001.635	1146938					
28		01-02-2026	0	1146938	0	1146938					
29											
30											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FillDown								
3		Category	Text								
4		Description	Fills down blank cells with the content above.								
5		Usage	=BI_FillDown(**Data**)								
6		Parameters	**Data** - A Column of data with blank cells in between.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Region	SalesMan	Sales	regionfilled						
13		Region1	S1	3079	Region1		**=BI_FillDown(B13:B27)**				
14			S2	1621	Region1						
15			S3	3132	Region1						
16			S4	3237	Region1						
17			S5	2055	Region1						
18		Region2	S1	1161	Region2						
19			S2	3666	Region2						
20			S3	2587	Region2						
21			S4	2153	Region2						
22			S5	1660	Region2						
23		Region3	S1	2715	Region3						
24			S2	2252	Region3						
25			S3	3464	Region3						
26			S4	2558	Region3						
27			S5	3100	Region3						
28											
29											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FinancialYear								
3		Category	Date								
4		Description	Returns the financial year (1st April to 31st March) of any date.								
5		Usage	=BI_FinancialYear([**Date**],[**YearEndMonth**],[**Format**])								
6		Parameters	**Date**: Date Serial Number to find the Financial Year (By default - Current System Date) **YearEndMonth**: Any month number from 1 to 12 (By default 3 - March) **Format**: Short or Long format 1 - Short Form, 2 - Half Full Form 3 - Full Form (By default 3)								
7											
8		Home									
9											
10		**Example:**									
11											
12		1	23-24	=BI_FinancialYear(,,B12)							
13		2	2023-24	=BI_FinancialYear(,,B13)							
14		3	2023-2024	=BI_FinancialYear(,,B14)							
15											
16											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FinancialYearEnd								
3		Category	Date								
4		Description	Returns the financial year end (31st March) of any date.								
5		Usage	=BI_FinancialYearEnd([**Date**],[**YearEndMonth**])								
6		Parameters	**Date**: Date Serial Number to find the Financial Year (By default - Current System Date) **YearEndMonth**: Any month number from 1 to 12 (By default 3 - March)								
7											
8		Home									
9											
10		**Example:**									
11											
12		31-03-2024									
13											
14		**=BI_FinancialYearEnd()**									
15											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FinancialYearStart								
3		Category	Date								
4		Description	Returns the financial year start (1st April) of any date.								
5		Usage	=BI_FinancialYearStart([**Date**],[**YearEndMonth**])								
6		Parameters	**Date**: Date Serial Number to find the Financial Year (By default - Current System Date) **YearEndMonth**: Any month number from 1 to 12 (By default 3 - March)								
7											
8		Home									
9											
10		**Example:**									
11											
12		01-04-2023									
13											
14		**=BI_FinancialYearStart()**									
15											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FindDuplicates								
3		Category	Audit								
4		Description	Returns duplicate values from a list								
5		Usage	=BI_FindDuplicates(**List**)								
6		Parameters	**List** - A column of data to find the duplicates								
7											
8		Home									
9											
10		**Example:**									
11											
12		InvoiceNumbers									
13		1		4		=BI_FindDuplicates(B13:B24)					
14		2		8							
15		3									
16		4									
17		5									
18		6									
19		4									
20		7									
21		8									
22		9									
23		8									
24		10									
25											
26											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FindMissingNumbers								
3		Category	Audit								
4		Description	Returns missing numbers from a list of numbers								
5		Usage	=BI_FindMissingNumbers(**List**)								
6		Parameters	**List** - A column of Numbers to find any number missing.								
7											
8		Home									
9											
10		**Example:**									
11											
12		InvoiceNumbers									
13		1001		1004		**=BI_FindMissingNumbers(B13:B20)**					
14		1002		1007							
15		1003		1008							
16		1005		1009							
17		1006									
18		1010									
19		1011									
20		1012									
21											
22											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FindOneOf								
3		Category	Audit								
4		Description	Returns the position of the first specified character value in a string. If the specified value is not found, 0 is returned. CaseWare IDEA inspiration.								
5		Usage	=BI_FindOneOf(**Text**,**Pattern**)								
6		Parameters	**Text** - Text to be searched in. **Pattern** - Any set of characters to verify								
7											
8		Home									
9											
10		**Example:**									
11											
12		ABC123XYZ									
13											
14		0123456789									
15											
16		4									
17											
18		**=BI_FindOneOf(B12,B14)**									
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FindUnique								
3		Category	Audit								
4		Description	Returns the values available only once (without duplicates) in a list. It is the opposite of Find Duplicates								
5		Usage	=BI_FindUnique(**List**)								
6		Parameters	**List** - A column of data to find only non repeated items.								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13		Products									
14		Camera									
15		Mobile Phone									
16		Camera									
17		Smart Watch									
18		Ring									
19		Plastic Bottle									
20		Plastic Bottle									
21											
22		Mobile Phone			Camera						
23		Smart Watch			Plastic Bottle						
24		Ring									
25											
26		=BI_FindUnique(B14:B20)			=BI_FindDuplicates(B14:B20)						
27											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_FormulaList								
3		Category	Audit								
4		Description	Returns the formula list in the specified range.								
5		Usage	=BI_FormulaList(**reference**)								
6		Parameters	**Reference** - Range Reference to identify and list the formulas.								
7											
8		Home									
9											
10		**Example:**									
11											
12		5									
13		4									
14		20									
15		25									
16		45									
17		2									
18		22.5									
19											
20		B14 =B12*B13									
21		B16 =B14+B15									
22		B18 =B16/B17									
23											
24		=BI_FormulaList(B12:B18)									
25											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Fuzzy								
3		Category	Audit								
4		Description	Returns the number of modifications required to change from first text to second text. This finds Levenshtein distance between two strings. Please see **https://en.wikipedia.org/wiki/Wagner%E2%80%93Fischer_algorithm** and **https://gist.github.com/ncalm/715a0507805ff1df95cde2a04a9709be#file-excel-lambda-lev-txt**								
5		Usage	=BI_Fuzzy(**Text1**,**Text2**)								
6		Parameters	**Text1** - a - First Text **Text2** - b - Second Text Note: Don't use other parameters as they are used for recursive calling.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Santhi									
13		Shanthi									
14											
15		1									
16											
17		=BI_Fuzzy(B12,B13)									
18											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_GEL1Test								
3		Category	Forensic Audit								
4		Description	Function to do the Gestalt Element Link Test 1. The connectedness principle can be used to identify link between data. For example, same salesman using more of the same discount rate. It creates a M x N combination of unique values in both the columns.								
5		Usage	=BI_GEL1Test(**AnalyzeData**,**RelatedData**)								
6		Parameters	**AnalyzeData** - Master Data to be Analyzed (example: Salesman) Don't Select any data having less replication example Customer Name. **RelatedData** - Any attribute of the data (example Discount Rate) Don't select any data having less replication.								
7											
8		Home									
9											
10		**Example:**									
11					=BI_GEL1Test(B13#,C13#)						
12											
13		Bookcases	0		Labels	0	239	364	0.656593		
14		Chairs	0		Storage	0	530	846	0.626478		
15		Labels	0		Art	0	498	796	0.625628		
16		Tables	0.45		Paper	0	857	1370	0.625547		
17		Storage	0.2		Supplies	0	117	190	0.615789		
18		Furnishings	0		Accessories	0	471	775	0.607742		
19		Art	0		Envelopes	0	152	254	0.598425		
20		Phones	0.2		Furnishings	0	571	957	0.596656		
21		Binders	0.2		Fasteners	0	128	217	0.589862		
22		Appliances	0		Appliances	0	271	466	0.581545		
23		Tables	0.2		Copiers	0.2	37	68	0.544118		
24		Phones	0.2		Phones	0.2	469	889	0.527559		
25		Paper	0.2		Fasteners	0.2	89	217	0.410138		
26		Binders	0.2		Chairs	0.2	250	617	0.405186		
27		Appliances	0.8		Envelopes	0.2	102	254	0.401575		
28		Binders	0.8		Accessories	0.2	304	775	0.392258		
29		Storage	0		Supplies	0.2	73	190	0.384211		
30		Storage	0		Binders	0.2	573	1523	0.376231		
31		Art	0		Paper	0.2	513	1370	0.374453		
32		Phones	0.2		Art	0.2	298	796	0.374372		
33		Binders	0.2		Storage	0.2	316	846	0.373522		
34		Art	0		Phones	0	311	889	0.349831		
35		Appliances	0		Labels	0.2	125	364	0.343407		
36		Chairs	0.3		Copiers	0	22	68	0.323529		
37		Tables	0		Machines	0.2	31	115	0.269565		
38		Binders	0.2		Bookcases	0	60	228	0.263158		
39		Accessories	0		Furnishings	0.2	248	957	0.259143		
40		Bookcases	0.5		Chairs	0.3	158	617	0.256078		
41		Binders	0.7		Machines	0	29	115	0.252174		
42		Furnishings	0.2		Binders	0.7	380	1523	0.249508		
43		Envelopes	0.2		Appliances	0.2	112	466	0.240343		
44		Art	0.2		Tables	0.4	75	319	0.23511		
45		Binders	0.7		Bookcases	0.15	52	228	0.22807		
46		Art	0.2		Tables	0	72	319	0.225705		
47		Paper	0.2		Tables	0.2	71	319	0.222571		

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_GEL2Test								
3		Category	Forensic Audit								
4		Description	Function to do the Gestalt Element Link Test 2. Further analysis of GEL 1 Test.								
5		Usage	=BI_GEL2Test(**AnalyzeData**,**RelatedData**,**DataItem**,[**Total**])								
6		Parameters	**AnalyzeData** - Data to be Analyzed (example: Discount Rate) Don't Select any data having less replication example sales value. **RelatedData** - Any attribute of the data (example Salesman) Don't select any data having less replication. **DataItem** - Item to be further analyzed from GEL1 Test **Total** - Any value column for total value: example Sales by the salesman (by default none).								
7											
8		Home									
9											
10		**Example:**	Select an Item:		**Binders**		**=BI_GEL2Test(B12#,C12#,E10, SalesTable[Sales])**				
11											
12		Bookcases	0			Count	Total Count	Value	TotalValue	Count/Tota	Value/TotalValue
13		Chairs	0		0.8	233	300	13581.22	16963.76	0.776667	0.800602
14		Labels	0		0.7	380	418	22559.39	40620.28	0.909091	0.555373
15		Tables	0.45		0.2	573	3657	85442.64	764594.4	0.156686	0.111749
16		Storage	0.2		0	337	4798	81829.48	1087908	0.070238	0.075217
17		Furnishings	0								
18		Art	0								
19		Phones	0.2								

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_GenerateRandom								
3		Category	Statistics								
4		Description	Function to generate **static** (not changing when recalulcated) random numbers between 0 and 1 using a Random Seed. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_GenerateRandom(**Seed**,**Size**)								
6		Parameters	**Seed** - Any random seed number for the starting point of the random algorithm. **Size** - Number of random numbers to be selected.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Seed	11111								
13		Size	20								
14											
15			0.795542		**=BI_GenerateRandom(C12,C13)**						
16			0.628798								
17			0.6929								
18			0.955805								
19			0.643055								
20			0.896382								
21			0.271201								
22			0.150404								
23			0.166849								
24			0.965269								
25			0.492494								
26			0.154132								
27			0.095084								
28			0.80963								
29			0.638174								
30			0.279285								
31			0.384412								
32			0.938576								
33			0.018854								
34			0.103869								
35											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_GenerateRandomBetween								
3		Category	Statistics								
4		Description	Function to generate **static** (not changing when recalulcated) random numbers between two numbers using a Random Seed. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_GenerateRandomBetween(Seed,Size,Bottom,Top,[Integer])								
6		Parameters	Seed - Any random seed number for the starting point of the random algorithm. Size - Number of random numbers to be selected. Bottom - Smallest Number the algorithm will return Top - Largest number the algorithm will return Integer - True for getting integer numbers instead of numbers with decimals (by default the value is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Seed	11111								
13		Size	20								
14		Bottom	1000								
15		Top	5000								
16											
17		**=BI_GenerateRandomBetween(C12,C13,C14,C15)**					**=BI_GenerateRandomBetween(C12,C13,C14,C15,TRUE)**				
18											
19		4182.169733					4182				
20		3515.190727					3515				
21		3771.598335					3771				
22		4823.218334					4823				
23		3572.219045					3572				
24		4585.529299					4585				
25		2084.804575					2084				
26		1601.617273					1601				
27		1667.395767					1667				
28		4861.075386					4861				
29		2969.974033					2969				
30		1616.527114					1616				
31		1380.336818					1380				
32		4238.519042					4238				
33		3552.694519					3552				
34		2117.141506					2117				
35		2537.646916					2537				
36		4754.305389					4754				
37		1075.416202					1075				
38		1415.475192					1415				
39											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Histogram								
3		Category	Statistics								
4		Description	Function to create Histogram analysis. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_Histogram(**Range**,[**BinStart**],[**BinIncrement**])								
6		Parameters	**Range** - A Column of data for grouping **BinStart** - Bucket/Bin Starting Number (by default minimum value) **BinIncrement** - Bin Gap (by default increment to get 10 buckets)								
7											
8		Home									
9											
10		**Example:**									
11											
12		**=BI_Histogram(SalesTable[Sales])**				=BI_Histogram(SalesTable[Sales],0,1000)					
13											
14		0	1			0-1000	9526				
15		0.444-2264.444	9875			1000-2000	328				
16		2264.444-4528.444	91			2000-3000	77				
17		4528.444-6792.444	13			3000-4000	23				
18		6792.444-9056.444	6			4000-5000	21				
19		9056.444-11320.444	5			5000-6000	4				
20		13584.444-15848.444	1			6000-7000	2				
21		15848.444-18112.444	1			7000-8000	1				
22		20376.444-22640.444	1			8000-9000	4				
23						9000-10000	3				
24						10000-11000	1				
25						11000-12000	1				
26						13000-14000	1				
27						17000-18000	1				
28						22000-23000	1				
29											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_IndianNum2Word								
3		Category	Indian								
4		Description	Returns the Indian spelling of a number. (Maximum 16 digits)								
5		Usage	=BI_IndianNum2Word(**Number**)								
6		Parameters	**Number** - Any integer number up to 16 digits								
7											
8		Home									
9											
10		**Example:**									
11											
12		1	One								
13		15	Fifteen								
14		1000	One Thousand								
15		15441	Fifteen Thousand Four Hundred and Forty One								
16		154546	One Lakh Fifty Four Thousand Five Hundred and Forty Six								
17		165465465	Sixteen Crores Fifty Four Lakhs Sixty Five Thousand Four Hundred and Sixty Five								
18											
19		**=BI_IndianNum2Word(B12)**									
20											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Insert								
3		Category	Text								
4		Description	Inserts a sub-string into Character field or string before a specified position.								
5		Usage	=BI_Insert(**Text**,**Position**,**InsertText**)								
6		Parameters	**Text** - Original Text **Position** - Position of the new text to be inserted. **InsertText** - New Text to be inserted								
7											
8		Home									
9											
10		**Example:**									
11											
12		Text	I Be Indian Excel Template								
13		Position	3								
14		InsertText	Love								
15											
16		I Love Be Indian Excel Template									
17											
18		**=BI_Insert(C12,C13,C14)**									
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_IsCharAtoZ								
3		Category	Text								
4		Description	Verifies whether the character is an alphabet from a to z.								
5		Usage	=BI_IsCharAtoZ(**Character**)								
6		Parameters	**Character** - Character to be verified. If there are more than one characters, first one will be verified								
7											
8		Home									
9											
10		**Example:**									
11											
12		A	TRUE		**=BI_IsCharAtoZ(B12)**						
13											
14		apple	TRUE								
15											
16		[]	FALSE								
17											
18											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_IsLeap								
3		Category	Date								
4		Description	Verifies whether the given year is leap.								
5		Usage	=BI_IsLeap([**Date**])								
6		Parameters	Date - Any date to find whether the year is leap or not. (By default it is the current system date.								
7											
8		Home									
9											
10		**Example:**									
11											
12		FALSE	**=BI_IsLeap()**								
13											
14		01-01-2000									
15		TRUE	**=BI_IsLeap(B14)**								
16											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_IsValidPAN								
3		Category	Audit								
4		Description	Verifies whether the given Indian Income Tax PAN number is valid or not.								
5		Usage	=BI_IsValidPAN(**PAN**)								
6		Parameters	**PAN** - Indian Income Tax Permanent Account Number								
7											
8		Home									
9											
10		**Example:**									
11											
12		ABCDE1234X	FALSE		**=BI_IsValidPAN(B12)**						
13											
14		AEOPR8387P	TRUE		**=BI_IsValidPAN(B14)**						
15											

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3		Function Name	BI_LuhnAlgorithm								
4		Category	Audit								
5		Description	Verifies whether the given Debit/Credit card number is valid according to Luhn Algorithm followed by many CC companies.								
6		Usage	=BI_LuhnAlgorithm(**CCNo**)								
7		Parameters									
8											
9		Home									
10											
11		**Example:**									
12											
13		1234567890123456		FALSE		**=BI_LuhnAlgorithm(B13)**					
14											
15		374245455400126		TRUE		**=BI_LuhnAlgorithm(B15)**					
16											
17		6250941006528599		TRUE		**=BI_LuhnAlgorithm(B17)**					
18											
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MonthlyCalendar								
3		Category	Date								
4		Description	Returns the Monthly Calendar of any date or current date in horizontal or vertical format.								
5		Usage	=BI_MonthlyCalendar([**Month**],[**Year**],[**Format**],[**Vertical**])								
6		Parameters	**Month** - Month Number between 1 to 12 (By default Current Month) **Year** - Year Number between 1900 to 9999 (By default Current Year) **Format** - Day Format - DDD - Short Form, DDDD - Long Form (By default short form) **Vertical** - True for vertical format of calendar (By default it is false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		அ்கமடோப்ர	2023								
13		ஞோ ியுற	திஙக்ள	செ்வவேர்ய	ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ினா			
14		1	2	3	4	5	6	7			
15		8	9	10	11	12	13	14			
16		15	16	17	18	19	20	21			
17		22	23	24	25	26	27	28			
18		29	30	31							
19											
20		**=BI_MonthlyCalendar()**									
21											
22		அ்கமடோப்ர	2023								
23		ஞோ ிய்றுற்கிகழதிஙக்ளிகழசெ்வவேர்ய்புத்னிகழலேிவயோழ்சவ்ளிள்கிகழ◌ிலேன்கிகழலே									
24		1	2	3	4	5	6	7			
25		8	9	10	11	12	13	14			
26		15	16	17	18	19	20	21			
27		22	23	24	25	26	27	28			
28		29	30	31							
29											
30		**=BI_MonthlyCalendar(,,"DDDD")**									
31											
32		அ்கமடோப்ர	2023								
33		ஞோ ியுற	1	8	15	22	29				
34		திஙக்ள	2	9	16	23	30				
35		செ்வவேர்ய	3	10	17	24	31				
36		புத்ன	4	11	18	25					
37		வியோழ்ன	5	12	19	26					
38		சவ்ளிள	6	13	20	27					
39		ெ◌ினா	7	14	21	28					
40											
41		**=BI_MonthlyCalendar(,,,TRUE)**									
42											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MovingAverage								
3		Category	Statistics								
4		Description	Returns the moving average calculation of data. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_MovingAverage(**Data**,[**Interval**],[**Labels**],[**StandardError**])								
6		Parameters	**Data** - A Column of Data for the Moving Average **Interval** - Interval for Moving Average (By default the value is 3) **Labels** - True if the first row of the data contains row labels. (By default the value is false) **StandardError** - True if required the standard error calculation also. (By default it is								
7											
8		Home									
9											
10		**Example:**									
11											
12		1999	#N/A	#N/A	#N/A						
13		1758	#N/A	#N/A	#N/A		**=BI_MovingAverage(B12#)**				
14		1178	1645	#N/A	#N/A						
15		1160	1365.333	#N/A	#N/A		**=BI_MovingAverage(B12#,5,,TRUE)**				
16		1812	1383.333	1581.4	#N/A						
17		1620	1530.667	1505.6	#N/A						
18		1934	1788.667	1540.8	#N/A						
19		1781	1778.333	1661.4	#N/A						
20		1292	1669	1687.8	279.9399						
21		1601	1558	1645.6	261.0152						
22		1443	1445.333	1610.2	266.6507						
23		1026	1356.667	1428.6	269.4415						
24		1077	1182	1287.8	280.4019						
25		1634	1245.667	1356.2	250.4561						
26		1219	1310	1279.8	251.1369						
27		1446	1433	1280.4	250.9248						
28		1033	1232.667	1281.8	207.1864						
29		1085	1188	1283.4	204.7227						
30		1862	1326.667	1329	288.608						
31		1174	1373.667	1320	294.6497						
32											
33											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MUS_Evaluate_OverStatement								
3		Category	Audit Sampling								
4		Description	Returns Monetary Unit Sampling evaluation of Over Statement from the Audited and Samples data.								
5		Usage	=BI_MUS_Evaluate_OverStatement(**Data**,**Sample**,**Audited**,**Tolerable**,**Expected**,**Risk**)								
6		Parameters	**Data** - A column of Data (population) **Sample** - Monetary Unit Sample created by BI_MUS_ExtractSample() **Audited** - Audited Values with over/understatement **Tolerable** - Tolerable Limit (Usually minimum double the amount of Expected) **Expected** - Expected Error Level (Can be 0) **Risk** - Alpha Level (Usually 0.05)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Data		=BI_MUS_Evaluate_OverStatement(B14#,D18:D47,E18:E47,0.15,0.02,0.05)							
13											
14		261.96	Population Size	9994							
15		731.94	Sample Size	30							
16		14.62	Monetary Unit	76573.36							
17		957.5775	RowID	Sample	Audited		#	Percentage of Misstatement	Upper Limit	Estimated Overstatement	
18		22.368	296	218.352	218.352		0	1	0.096	220531.3	
19		48.86	586	406.368	**450**		1	0.107371	0.119	29351.58	
20		7.28	970	501.81	501.81		2	0.05851	0.149	20026.89	
21		907.152	1275	388.43	388.43		3	0.018535	0.177	7536.616	
22		18.504	1660	88.752	88.752		Total			277446.4	
23		114.9	2014	350.98	350.98		Tolerable			344580.1	
24		1706.184	2409	359.976	359.976		Result			Accept	
25		911.424	2625	2399.6	**2540**						
26		15.552	2861	1628.82	1628.82						
27		407.976	3246	49.08	49.08						
28		68.81	3591	1443.96	1443.96						
29		2.544	3987	2591.56	2591.56						
30		665.88	4267	956.6648	956.6648						
31		55.5	4620	2879.952	2879.952						
32		8.56	5023	772.47	**700**						
33		213.48	5386	545.916	545.916						
34		22.72	5772	1158.12	1158.12						
35		19.46	6100	2888.127	2888.127						
36		60.34	6426	8399.976	8399.976						
37		71.372	6682	146.82	146.82						
38		1044.63	6958	51.75	51.75						
39		11.648	7362	194.352	194.352						
40		90.57	7667	5399.91	**5500**						
41		3083.43	7998	1673.184	1673.184						
42		9.618	8260	368.91	368.91						
43		124.2	8623	39.808	39.808						
44		3.264	8987	95.97	**80**						
45		86.304	9288	457.485	457.485						
46		6.858	9644	481.568	481.568						
47		15.76	9994	243.16	243.16						
48		29.472									
49		1097.544									

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MUS_Evaluate_UnderStatement								
3		Category	Audit Sampling								
4		Description	Returns Monetary Unit Sampling evaluation of Under Statement from the Audited and Samples data.								
5		Usage	=BI_MUS_Evaluate_UnderStatement(**Data**,**Sample**,**Audited**,**Tolerable**,**Expected**,**Risk**)								
6		Parameters	**Data** - A column of Data (population) **Sample** - Monetary Unit Sample created by BI_MUS_ExtractSample() **Audited** - Audited Values with over/understatement **Tolerable** - Tolerable Limit (Usually minimum double the amount of Expected) **Expected** - Expected Error Level (Can be 0) **Risk** - Alpha Level (Usually 0.05)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Data		=BI_MUS_Evaluate_UnderStatement(B14#,D18:D47,E18:E47,0.15,0.02,0.05)							
13											
14		261.96	Population	Size 9994							
15		731.94	Sample Size	30							
16		14.62	Monetary Unit	76573.36							
17		957.5775	RowID	Sample	Audited		#	Percentage of Misstatement	Upper Limit	Estimated Understatement	
18		22.368	296	218.352	218.352		0	1	0.096	220531.3	
19		48.86	586	406.368	**450**		1	0.166406	0.119	45489.94	
20		7.28	970	501.81	501.81		2	0.093816	0.149	32111.6	
21		907.152	1275	388.43	388.43		Total			298132.8	
22		18.504	1660	88.752	88.752		Tolerable			344580.1	
23		114.9	2014	350.98	350.98		Result			Accept	
24		1706.184	2409	359.976	359.976						
25		911.424	2625	2399.6	**2540**						
26		15.552	2861	1628.82	1628.82						
27		407.976	3246	49.08	49.08						
28		68.81	3591	1443.96	1443.96						
29		2.544	3987	2591.56	2591.56						
30		665.88	4267	956.6648	956.6648						
31		55.5	4620	2879.952	2879.952						
32		8.56	5023	772.47	**700**						
33		213.48	5386	545.916	545.916						
34		22.72	5772	1158.12	1158.12						
35		19.46	6100	2888.127	2888.127						
36		60.34	6426	8399.976	8399.976						
37		71.372	6682	146.82	146.82						
38		1044.63	6958	51.75	51.75						
39		11.648	7362	194.352	194.352						
40		90.57	7667	5399.91	**5500**						
41		3083.43	7998	1673.184	1673.184						
42		9.618	8260	368.91	368.91						
43		124.2	8623	39.808	39.808						
44		3.264	8987	95.97	**80**						
45		86.304	9288	457.485	457.485						
46		6.858	9644	481.568	481.568						
47		15.76	9994	243.16	243.16						
48		29.472									
49		1097.544									

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MUS_ExtractSample								
3		Category	Audit Sampling								
4		Description	Returns Monetary Unit Sampling Samples from Data.								
5		Usage	=BI_MUS_ExtractSample(**Data**,**Tolerable**,**Expected**,**Risk**,[**OnlySamples**])								
6		Parameters	**Data** - A column of Data (population) **Tolerable** - Tolerable Limit (Usually minimum double the amount of Expected) **Expected** - Expected Error Level (Can be 0) **Risk** - Alpha Level (Usually 0.05) **OnlySamples** - True for only the samples to be returned (by default the value is false)								
7											
8		Home									
9											
10		**Example:**		**=BI_MUS_ExtractSample(B14#,0.15,0.02,0.05,TRUE)**							
11											
12		Data		**=BI_MUS_ExtractSample(B14#,0.15,0.02,0.05)**							
13											
14		261.96	Population Size 9994			Population Size 9994					
15		731.94	Sample Size	30		Sample Size	30				
16		14.62	Monetary Un	76573.36		Monetary Un	76573.36				
17		957.5775	RowID	Sample		RowID	Sample	RunningTotal	MU Hit		
18		22.368	296	218.352		296	218.352	76682.78	76573.36		
19		48.86	586	406.368		586	406.368	153428.8	153146.7		
20		7.28	970	501.81		970	501.81	230032.5	229720.1		
21		907.152	1275	388.43		1275	388.43	306339.4	306293.4		
22		18.504	1660	88.752		1660	88.752	382886.3	382866.8		
23		114.9	2014	350.98		2014	350.98	459731.7	459440.2		
24		1706.184	2409	359.976		2409	359.976	536245.9	536013.5		
25		911.424	2625	2399.6		2625	2399.6	613904.8	612586.9		
26		15.552	2861	1628.82		2861	1628.82	690083.2	689160.3		
27		407.976	3246	49.08		3246	49.08	765776.6	765733.6		
28		68.81	3591	1443.96		3591	1443.96	843382.5	842307		
29		2.544	3987	2591.56		3987	2591.56	919801.3	918880.3		
30		665.88	4267	956.6648		4267	956.6648	996317.7	995453.7		
31		55.5	4620	2879.952		4620	2879.952	1072932	1072027		
32		8.56	5023	772.47		5023	772.47	1148869	1148600		
33		213.48	5386	545.916		5386	545.916	1225716	1225174		
34		22.72	5772	1158.12		5772	1158.12	1302735	1301747		
35		19.46	6100	2888.127		6100	2888.127	1380457	1378321		
36		60.34	6426	8399.976		6426	8399.976	1459763	1454894		
37		71.372	6682	146.82		6682	146.82	1531536	1531467		
38		1044.63	6958	51.75		6958	51.75	1608074	1608041		
39		11.648	7362	194.352		7362	194.352	1684687	1684614		
40		90.57	7667	5399.91		7667	5399.91	1764385	1761187		
41		3083.43	7998	1673.184		7998	1673.184	1838518	1837761		
42		9.618	8260	368.91		8260	368.91	1914400	1914334		
43		124.2	8623	39.808		8623	39.808	1990938	1990907		
44		3.264	8987	95.97		8987	95.97	2067527	2067481		
45		86.304	9288	457.485		9288	457.485	2144479	2144054		
46		6.858	9644	481.568		9644	481.568	2220930	2220627		
47		15.76	9994	243.16		9994	243.16	2297201	2297201		
48		29.472									
49		1097.544									
50		190.92									

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_MUS_SampleSize								
3		Category	Audit Sampling								
4		Description	Returns Monetary Unit Sample size.								
5		Usage	=BI_MUS_SampleSize(**Tolerable**,**Expected**,**Risk**,[**Start**])								
6		Parameters	**Tolerable** - Tolerable Limit (Usually minimum double the amount of Expected) **Expected** - Expected Error Level (Can be 0) **Risk** - Alpha Level (Usually 0.05) **Start** - Starting Sample Size (By default 1)								
7											
8		Home									
9											
10		**Example:**									
11											
12		30									
13											
14		**=BI_MUS_SampleSize(0.15,0.02,0.05)**									
15											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Networkdays.Indian								
3		Category	Indian								
4		Description	Returns the number of whole work days between two dates considering Indian banking holidays (second and fourth Saturdays as holidays).								
5		Usage	=BI_NetworkDays.Indian(**StartDate**,**EndDate**,**Holidays**)								
6		Parameters	**StartDate** - Starting Date of the Project **EndDate** - Ending Date/Deadline of the Project **Holidays** - List of Holidays								
7											
8		Home									
9											
10		**Example:**									
11											
12		Start Date	15-10-2023								
13		Ending Date	15-02-2024								
14											
15		Holidays									
16		12-11-2023									
17		25-12-2023									
18		31-12-2023									
19		01-01-2024									
20		14-01-2024									
21		26-01-2024									
22											
23		95									
24											
25		**=BI_NetworkDays.Indian(C12,C13,B16:B21)**									
26											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Num2Word								
3		Category	Text								
4		Description	Returns the English spelling of a Number								
5		Usage	=BI_Num2Word(**Number**)								
6		Parameters	Number - An Integer Number up to Trillion								
7											
8		Home									
9											
10		**Example:**									
11											
12		1	One								
13		15545221	Fifteen Million Five Hundred and Forty Five Thousand Two Hundred and Twenty One								
14		21656565	Twenty One Million Six Hundred and Fifty Six Thousand Five Hundred and Sixty Five								
15		65465465494	Sixty Five Billion Four Hundred and Sixty Five Million Four Hundred and Sixty Five Thousand Four Hundred and Ninety Four								
16		6.54655E+11	Six Hundred and Fifty Four Billion Six Hundred and Fifty Four Million Six Hundred and Fifty Four Thousand Six Hundred and Fifty Eight								
17											
18			**=BI_Num2Word(B12)**								
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_OhlsonsOScore								
3		Category	Forensic Audit								
4		Description	Calculates Ohlsons O Score. It is used for predicting bankruptcy is a multi-factor financial formula postulated in 1980 by Dr. James Ohlson of the New York University								
5		Usage	=BI_OhlsonsOScore(**Data**)								
6		Parameters	**Data** - a Column of Data for calculating Ohlsons O Score in the order of the variables as listed in BI_OhlsonOScoreVariables()								
7											
8		Home									
9											
10		**Example:**									
11											
12		Total Assets			1000000						
13		Gross National Product Price Level Index			156						
14		Total Liabilities			750000						
15		Working Capital			100000						
16		Current Liabilities			300000						
17		Current Assets			400000						
18		LastYear Net Income			80000						
19		Current Year Net Income			90000						
20		Funds From Operations			96000						
21											
22		Ohlsons O Score	-2.68781								
23		Verdict	The firm is not likely to default within two years.								
24											
25		**=BI_OhlsonsOScore(E12:E20)**									
26											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_OhlsonsOScoreVariables								
3		Category	Forensic Audit								
4		Description	Lists out Ohlsons O Score Variables.								
5		Usage	=BI_OhlsonsOScoreVariables()								
6		Parameters	Nil								
7											
8		Home									
9											
10		**Example:**									
11											
12		Total Assets									
13		Gross National Product Price Level Index									
14		Total Liabilities									
15		Working Capital									
16		Current Liabilities									
17		Current Assets									
18		LastYear Net Income									
19		Current Year Net Income									
20		Funds From Operations									
21											
22		**=BI_OhlsonsOScoreVariables()**									
23											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_PaybackPeriod								
3		Category	Financial								
4		Description	Returns the payback period of an investment.								
5		Usage	=BI_PaybackPeriod(**CashFlows**)								
6		Parameters	CashFlows - Inflows are positive and Outflows are negative.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Period	Cashflow								
13		0	-1000000								
14		1	150000								
15		2	150000								
16		3	200000								
17		4	250000								
18		5	200000								
19		6	250000								
20		7	250000								
21		8	250000								
22											
23		**5.2**									
24											
25		**=BI_PaybackPeriod(C13:C21)**									
26											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_PiotroskiFScore								
3		Category	Forensic Audit								
4		Description	Calculates Piotroski F Score. It is a discrete score between zero and nine that reflects nine criteria used to determine the strength of a firm's financial position.								
5		Usage	=BI_PiotroskiFScore(**LastYearData**,**CurrentYearData**)								
6		Parameters	**LastYearData** - Last Year Data in the order of variables as listed in BI_PiotroskiFScoreVariables() **CurrentYearData** - Current Year Data in the order of variables as listed in BI_PiotroskiFScoreVariables()								
7											
8		Home									
9											
10		**Example:**									
11			last year	CurrentYear							
12		Net Income	150000	160000							
13		Opening Total Assets	900000	950000							
14		Closing Total Assets	950000	980000							
15		Cashflow from Operations	180000	185000							
16		Long Term Debt	300000	350000							
17		Current Assets	200000	225000							
18		Current Liabilities	140000	148000							
19		Common Equity	150000	150000							
20		Net Sales	600000	700000							
21		GP Ratio	20%	20.50%							
22											
23		Positive Net Income	1								
24		Positive Cash Flow	1								
25		Change in Return on Assets	1								
26		Cash Flow over Net Income	1								
27		Leverage - Long term debt over Average Assets	0								
28		Liquidity - Current Ratio	1								
29		Total Equity	1								
30		GP Ratio	1								
31		Asset Turnover	1								
32		Total Score	8								
33		Verdict	8 Out of 9 - Strong!								
34											
35		**=BI_PiotroskiFScore(C12:C21,D12:D21)**									
36											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_PiotroskiFScoreVariables								
3		Category	Forensic Audit								
4		Description	Lists out Piotroski F Score variables to make it easy to enter two years data for calculating Piotroski F Score.								
5		Usage	=BI_PiotroskiFScoreVariables()								
6		Parameters	Nil								
7											
8		Home									
9											
10		**Example:**									
11											
12		Net Income									
13		Opening Total Assets									
14		Closing Total Assets									
15		Cashflow from Operations									
16		Long Term Debt									
17		Current Assets									
18		Current Liabilities									
19		Common Equity									
20		Net Sales									
21		GP Ratio									
22											
23		**=BI_PiotroskiFScoreVariables()**									
24											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Pivot								
3		Category	Data								
4		Description	Creates a Pivot Table from Data.								
5		Usage	=BI_Pivot(**Data**,**Value**,**Row**,**Col**,[**Function**])								
6		Parameters	**Data** - TableData or Array **Value** - Column Index of the Value (Example: Sales) **Row** - Column Index of the Row Labels (Example: Sub-Category) **Col** - Column Index of the Column Labels (Example: Segment) **Function** - Summary Function to be evaluated as in Subtotal() function. (By default the value is 9 (sum))								
7											
8		Home									
9											
10		**Example:**									
11											
12			Consumer	Corporate	Home Office						
13		Accessories	87105.24	48190.56	32084.52						
14		Appliances	52819.58	36588.68	18123.9						
15		Art	14251.93	8590.448	4276.414						
16		Binders	118161	51560.31	33691.41						
17		Bookcases	68632.73	34005.92	12241.34						
18		Chairs	172862.7	99140.88	56445.48						
19		Copiers	69819.07	46829.39	32879.57						
20		Envelopes	7771.146	5942.67	2762.586						
21		Fasteners	1680.942	783.29	560.048						
22		Furnishings	49620.05	25001.27	17083.85						
23		Labels	6709.262	4101.646	1675.404						
24		Machines	79542.83	60276.76	49419.05						
25		Paper	36324.37	23883.13	18271.71						
26		Phones	169932.8	91153.41	68920.88						
27		Storage	100492.4	79791	43560.21						
28		Supplies	25741.5	19435.28	1496.758						
29		Tables	99933.8	70871.72	36160.02						
30											
31											
32		**=BI_Pivot(SalesTable,12,10,7)**									
33											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Quarter								
3		Category	Date								
4		Description	Returns the Quarter number for a date.								
5		Usage	=BI_Quarter([**Date**],[**YearEndMonth**])								
6		Parameters	**Date** - Date for the which the quarter number to be calculated (By default the value is current system date) **YearEndMonth** - Financial Year End Month (By default the value is 3 - March)								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13		01-03-2024									
14											
15		3	**=BI_Quarter()**								
16											
17		4	**=BI_Quarter(B13)**								
18											
19		1	**=BI_Quarter(B13,12)**								
20											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_QuarterEnd								
3		Category	Date								
4		Description	Returns the Quarter ending date.								
5		Usage	=BI_QuarterEnd([**Date**],[**YearEndMonth**])								
6		Parameters	**Date** - Date Serial Number to find the Quarter End (By default the value is the current system date) **YearEndMonth** - Month number from 1 to 12 as Year End Month. (By default the value is 3 - March for financial year)								
7											
8		Home									
9											
10		**Example:**									
11											
12		31-12-2023	**=BI_QuarterEnd()**								
13											
14		31-10-2023	**=BI_QuarterEnd(,10)**								
15											
16											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_QuarterStart								
3		Category	Date								
4		Description	Returns the Quarter starting date.								
5		Usage	=BI_QuarterStart([**Date**],[**YearEndMonth**])								
6		Parameters	**Date** - Date Serial Number to find the Quarter Start (By default the value is the current system date) **YearEndMonth** - Month number from 1 to 12 as Year End Month. (By default the value is 3 - March for financial year)								
7											
8		Home									
9											
10		**Example:**									
11											
12		01-10-2023	**=BI_QuarterStart()**								
13											
14		01-08-2023	**=BI_QuarterStart(,10)**								
15											
16											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Rank.Dense								
3		Category	Statistics								
4		Description	Returns the Rank without any missing rank when there is a tie.								
5		Usage	=BI_Rank.Dense(**Number**,**Ref**,[**Order**])								
6		Parameters	**Number** - Number to be ranked. **Ref** - Range of Numbers among the rank is calculated. **Order**: -1 for descending order, 1 for ascending order (by default the value is descending order)								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13		7	6	7							
14		1	12	1		=BI_Rank.Dense(B13#,B13#)					
15		3	10	3							
16		2	11	2		=BI_Rank.Dense(B13#,B13#,1)					
17		5	8	5							
18		14	1	12							
19		2	11	2							
20		14	1	12							
21		4	9	4							
22		9	5	8							
23		12	3	10							
24		1	12	1							
25		12	3	10							
26		3	10	3							
27		13	2	11							
28		9	5	8							
29		5	8	5							
30		1	12	1							
31		6	7	6							
32		10	4	9							
33											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Regression								
3		Category	Statistics								
4		Description	Calculates the Linear Regression. This is one of the Data Analysis Toolpak features.								
5		Usage	=BI_Regression(**KnownYs**,**KnownXs**,[**Labels**],[**Confidence**])								
6		Parameters	**KnownYs** - Known Y Values **KnownXs** - Known X Values **Labels** - True for data with first row containing Row Labels (By default false) **Confidence** - Confidence Interval (By default 0.95)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Company1	Company2	Company3	Company4						
13		1197	1065	1174	1073						
14		1051	1131	1113	1141						
15		1070	1066	1074	1129						
16		1164	1074	1144	1102		**=BI_Regression(B12:B32,C12:E32,TRUE,0.9)**				
17		1105	1078	1064	1191						
18		1040	1066	1017	1084						
19		1032	1110	1057	1070						
20		1080	1095	1183	1134						
21		1062	1183	1026	1135						
22		1075	1132	1070	1018						
23		1192	1124	1157	1113						
24		1007	1074	1025	1161						
25		1124	1096	1109	1089						
26		1026	1021	1168	1161						
27		1074	1023	1159	1080						
28		1151	1195	1160	1200						
29		1093	1023	1055	1062						
30		1191	1019	1043	1097						
31		1188	1118	1035	1200						
32		1159	1102	1026	1083						
33											
34		Summary Output									
35											
36		Regression Statistics									
37		Multiple R	0.22566965								
38		R Square	0.05092679								
39		Adjusted R Square	-0.1270244								
40		Standard Error	65.8907562								
41		Observations	20								
42											
43		ANOVA									
44			df	SS	MS	F	Significance F				
45		Regression	3	3727.48199	1242.494	0.286184	0.834666				
46		Residual	16	69465.468	4341.59175						
47		Total	19	73192.95							
48											
49			Coefficients	Standard Error	t Stat	P-Value	Lower 0.9	Upper 0.9	Lower 0.9	Upper 0.9	
50		Intercept	763.336518	502.755668	1.51830515	0.148448	-114.416	1641.089	-114.416	1641.089	
51		Company2	0.1213332	0.31707003	0.38267003	0.707	-0.43223	0.674901	-0.43223	0.674901	
52		Company3	0.22228406	0.25725595	0.86405801	0.400325	-0.22685	0.671423	-0.22685	0.671423	
53		Company4	-0.0308693	0.31898123	-0.0967746	#NUM!	-0.58777	0.526035	-0.58777	0.526035	
54											

	A	B	C	D	E	F	G	H	I	J	K
55		Observation	Predicted Y	Residuals							
56		1	1120.39514	76.6048646							
57		2	1112.74469	-61.744689							
58		3	1096.55938	-26.559383							
59		4	1113.9234	50.0765965							
60		5	1093.87865	11.1213535							
61		6	1085.27831	-45.278309							
62		7	1099.9405	-67.940502							
63		8	1124.15266	-44.152662							
64		9	1099.90052	-37.900517							
65		10	1107.10473	-32.104727							
66		11	1122.54019	69.4598057							
67		12	1085.65031	-78.650313							
68		13	1109.21409	14.7859076							
69		14	1111.00627	-85.006274							
70		15	1111.7488	-37.748795							
71		16	1129.13608	21.8639228							
72		17	1089.1869	3.81310017							
73		18	1084.95373	106.046266							
74		19	1092.00791	95.9920867							
75		20	1091.67773	67.3222698							
76											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_RelativeSizeFactor								
3		Category	Forensic Audit								
4		Description	Returns the Relative Size Factor of given data on any attribute. The result will be shown in descending order of the Relative Size Factor.								
5		Usage	=BI_RelativeSizeFactor(**Categories**,**RowValue**,**First**,**Second**)								
6		Parameters	**Categories** - A Column of categories data (Example: CustomerID) **RowValue** - A Column of value data (Example: Sales) **First** - First Largest number to be in the nominator (usually 1. Should be >= 1) **Second** - Second Largest number to be in the denominator (usually 2. Should be > First)								
7											
8		Home									
9											
10		**Example:**									
11			**=BI_RelativeSizeFactor(SalesTable[Customer ID],SalesTable[Sales],1,2)**								
12											
13		Category	First Value	Second Val	Relative Size Factor						
14		IM-15055	2799.96	40.08	69.85928						
15		RB-19360	13999.96	263.88	53.05427						
16		LH-16750	271.96	8.896	30.57104						
17		SM-20320	22638.48	821.3	27.5642						
18		MH-17620	1145.6	44.46	25.76698						
19		RH-19555	704.76	27.396	25.72492						
20		TC-20980	17499.95	735.98	23.77775						
21		TB-21190	767.214	34.92	21.97062						
22		EK-13795	1198.33	56.3	21.28472						
23		BO-11425	124.75	6.08	20.51809						
24		PC-19000	1004.976	49.792	20.18348						
25		CS-11860	479.984	25.344	18.93876						
26		CP-12085	4164.05	238.152	17.48484						
27		JL-15130	934.956	59.24	15.78251						
28		PH-18790	683.952	45.696	14.96744						
29		DI-13600	658.746	44.4	14.83662						
30		PB-19105	604.656	40.776	14.82872						
31		BP-11230	1022.97	70.98	14.41209						
32		PJ-18835	826.11	59.9	13.79149						
33		TP-21415	4416.174	322.59	13.68974						
34		GH-14425	3266.376	247.104	13.21863						
35		SF-20200	2665.62	207.24	12.86248						
36		AS-10285	2678.94	209.88	12.76415						
37		DP-13390	4535.976	359.058	12.63299						
38		SR-20425	2549.985	212.94	11.97513						
39		BT-11485	355.96	30.336	11.73391						
40		AT-10735	2625.12	223.96	11.72138						
41		HZ-14950	723.92	65.424	11.06505						
42		SW-20755	579.136	52.792	10.97015						
43		LW-17215	3785.292	359.97	10.51558						
44		SC-20845	237.096	22.752	10.42089						
45		BS-11365	9099.93	887.103	10.25803						
46		PC-18745	1573.488	155.82	10.09811						
47		CS-12505	4499.985	456.588	9.85568						
48		HL-15040	10499.97	1067.94	9.831985						
49		CB-12415	1439.976	146.82	9.807765						

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ReverseText								
3		Category	Text								
4		Description	Returns the text in reverse order.								
5		Usage	=BI_ReverseText(**Text**)								
6		Parameters	**Text** - Text to be reversed.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Malayalam									
13											
14		malayalaM		**=BI_ReverseText(B12)**							
15											
16		Reverse this text!									
17											
18		!txet siht esreveR		**=BI_ReverseText(B16)**							
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_RoundNumbersTest								
3		Category	Forensic Audit								
4		Description	Returns the Round numbers test in data.								
5		Usage	=BI_RoundNumbersTest(**Data**,**ColumnIndex**,**Digits**)								
6		Parameters	**Data** - TableData or Array **ColumnIndex** - Column Number to be tested (Example: Sales) **Digits** - Number of Digits from the last (Usually 2 or 3)								
7											
8		Home									
9											
10		**Example:**		**=BI_RoundNumbersTest(CHOOSECOLS(SalesTable,2,12),2,2)**							
11											
12		CA-2016-106894	100								
13		CA-2016-169957	100								
14		CA-2016-121748	100								
15											
16		Round Numbers Count	3								
17		Total Count	9994								
18		Round Amount Total	300								
19		Total Amount	2297201								
20		Count %	0.0003								
21		Amount %	0.000131								
22											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_RowOccurence								
3		Category	Array								
4		Description	Returns how many times the same complete row or selected columns occurs in the data.								
5		Usage	=BI_RowOccurence(**Data**,[**SelectedColumns**])								
6		Parameters									
7											
8		Home									
9											
10		**Example:**									
11											
12		RollNo	Name	Mark1	Mark2						
13		1	A	4	79	1	1				
14		2	B	77	55	1	1				
15		3	C	89	44	2	3				
16		4	D	80	25	1	1				
17		3	C	89	44	2	3				
18		6	F	66	41	1	1				
19		7	C	89	44	1	3				
20											
21											
22		**=BI_RowOccurence(B13:E19)**									
23											
24		**=BI_RowOccurence(B13:E19,{2,3,4})**									
25											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_RunningTotal								
3		Category	Data								
4		Description	Returns running total (cumulative total) of a set of data.								
5		Usage	=BI_RunningTotal(**Data**)								
6		Parameters	**Data** - A Column of Data for the running total.								
7											
8		Home									
9											
10		**Example:**									
11											
12		261.96	261.96		**=BI_RunningTotal(B12#)**						
13		731.94	993.9								
14		14.62	1008.52								
15		957.5775	1966.098								
16		22.368	1988.466								
17		48.86	2037.326								
18		7.28	2044.606								
19		907.152	2951.758								
20		18.504	2970.262								
21		114.9	3085.162								
22		1706.184	4791.346								
23		911.424	5702.77								
24		15.552	5718.322								
25		407.976	6126.298								
26		68.81	6195.108								
27		2.544	6197.652								
28		665.88	6863.532								
29		55.5	6919.032								
30		8.56	6927.592								
31		213.48	7141.072								
32		22.72	7163.792								
33		19.46	7183.252								
34		60.34	7243.592								
35		71.372	7314.964								
36		1044.63	8359.594								
37		11.648	8371.242								
38		90.57	8461.812								
39		3083.43	11545.24								
40		9.618	11554.86								

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_RupeeSymbol								
3		Category	Indian								
4		Description	Returns the Indian Rupee Symbol.								
5		Usage	=BI_RupeeSymbol()								
6		Parameters	Nil								
7											
8		Home									
9											
10		**Example:**									
11											
12		₹									
13											
14		**=BI_RupeeSymbol()**									
15											
16											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_SameSameDifferentTest								
3		Category	Forensic Audit								
4		Description	Returns Same values in specified columns and different value in third column. Useful in forensic audit to identify duplicates in some columns but different in another column. Example Same Address, Different Employee ID								
5		Usage	=BI_SameSameDifferentTest(**Data**,**TestColumns**,**DifferentColumn**)								
6		Parameters	**Data** - TableData or Array **TestColumns** - Columns where duplicates to be found **DifferentColumn** - Column Index where the data to be unique								
7											
8		Home									
9											
10		**Example:**									
11											
12		Let us find same city names in different states.									
13											
14		Apple Valley	Minnesota		**=SORT(BI_SameSameDifferentTest(Place,3,4))**						
15		Apple Valley	California								
16		Arlington	Virginia								
17		Arlington	Texas								
18		Auburn	New York								
19		Auburn	Washington								
20		Auburn	Alabama								
21		Aurora	Colorado								
22		Aurora	Illinois								
23		Bloomington	Illinois								
24		Bloomington	Indiana								
25		Bowling Green	Kentucky								
26		Bowling Green	Ohio								
27		Bristol	Tennessee								
28		Bristol	Connecticut								
29		Burlington	North Carolina								
30		Burlington	Iowa								
31		Burlington	Vermont								
32		Columbia	South Carolina								
33		Columbia	Tennessee								
34		Columbia	Maryland								
35		Columbia	Missouri								
36		Columbus	Ohio								
37		Columbus	Georgia								
38		Columbus	Indiana								
39		Concord	North Carolina								
40		Concord	California								
41		Concord	New Hampshire								
42		Danville	California								
43		Danville	Illinois								
44		Decatur	Alabama								
45		Decatur	Illinois								
46		Des Moines	Washington								
47		Des Moines	Iowa								
48		Dover	Delaware								

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_SameSameSameTest								
3		Category	Forensic Audit								
4		Description	Returns Same values in the given columns (like invoice number, supplier id, employee id). Useful in forensic audit.								
5		Usage	=BI_SameSameSameTest(**Data**,**TestColumns**)								
6		Parameters	**Data** - TableData or Array **TestColumns** - An array of Column Index Numbers to identify duplicates								
7											
8		Home									
9											
10		**Example:**		**=BI_SameSameSameTest(CHOOSECOLS(SalesTable,1,2,9,12,13),{2,3})**							
11											
12		Let us find order ID with the same product ID duplicated.									
13											
14											
15		351	CA-2016-129714	OFF-PA-10001970	24.56	2					
16		353	CA-2016-129714	OFF-PA-10001970	49.12	4					
17		431	US-2016-123750	TEC-AC-10004659	408.744	7					
18		432	US-2016-123750	TEC-AC-10004659	291.96	5					
19		1301	CA-2016-137043	FUR-FU-10003664	572.76	6					
20		1302	CA-2016-137043	FUR-FU-10003664	286.38	3					
21		3184	CA-2017-152912	OFF-ST-10003208	1633.14	9					
22		3185	CA-2017-152912	OFF-ST-10003208	544.38	3					
23		3406	US-2014-150119	FUR-CH-10002965	281.372	2					
24		3407	US-2014-150119	FUR-CH-10002965	281.372	2					
25		6499	CA-2015-103135	OFF-BI-10000069	135.09	9					
26		6501	CA-2015-103135	OFF-BI-10000069	90.06	6					
27		7882	CA-2017-118017	TEC-AC-10002006	76.752	6					
28		7883	CA-2017-118017	TEC-AC-10002006	102.336	8					
29		9169	CA-2016-140571	OFF-PA-10001954	319.76	14					
30		9170	CA-2016-140571	OFF-PA-10001954	45.68	2					
31											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_SLNSchedule								
3		Category	Financial								
4		Description	Returns Straight Line Method of Depreciation Schedule.								
5		Usage	=BI_SLNSchedule(**Cost**,**Life**,**Scrap**)								
6		Parameters	**Cost** - Cost of the Asset **Life** - Life of the Asset **Scrap** - Residual value at the end of the life								
7											
8		Home									
9											
10		**Example:**									
11											
12		Cost	1000000								
13		Life	10		**=BI_SLNSchedule(C12,C13,C14)**						
14		Scrap	100000								
15											
16		SLN Rate :	0.09								
17		Year	Op.Balance	Depreciation	Cl.Balance						
18		1	1000000	90000	910000						
19		2	910000	90000	820000						
20		3	820000	90000	730000						
21		4	730000	90000	640000						
22		5	640000	90000	550000						
23		6	550000	90000	460000						
24		7	460000	90000	370000						
25		8	370000	90000	280000						
26		9	280000	90000	190000						
27		10	190000	90000	100000						
28		Total		900000							
29											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_SubsetNumberDuplicationTest								
3		Category	Forensic Audit								
4		Description	Calculates the Subset Number Duplication Test. Example: Same Salesman and Same amount of Sales								
5		Usage	=BI_SubsetNumberDuplicationTest(**CategoryData**,**SubsetData**)								
6		Parameters	**CategoryData** - A column of data to be tested (Example Salesman) **SubsetData** - A column of data to be verified with number of duplication (Example Sales)								
7											
8		Home									
9											
10		**Example:**									
11			**=BI_SubsetNumberDuplicationTest(SalesTable[Customer ID],SalesTable[Sales])**								
12											
13		SJ-20215	0.055556								
14		GM-14695	0.047337								
15		MK-18160	0.047337								
16		AD-10180	0.040816								
17		AS-10630	0.040816								
18		YS-21880	0.040816								
19		JP-16135	0.035556								
20		CT-11995	0.02								
21		JG-15160	0.02								
22		CC-12430	0.018141								
23		MC-18100	0.018141								
24		DM-13015	0.016529								
25		JE-15715	0.013889								
26		RB-19465	0.0128								
27		JA-15970	0.011834								
28		LB-16795	0.011834								
29		SP-20860	0.011834								
30		DK-12835	0.009512								
31		Dp-13240	0.008325								
32		AP-10915	0.007346								
33		EP-13915	0.007346								
34		ZC-21910	0.007346								
35		EH-13765	0.00692								
36		JD-15895	0.00692								
37		JL-15835	0.006173								
38											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Summary								
3		Category	Data								
4		Description	Returns Summary of data according to any attribute.								
5		Usage	=BI_Summary(**Data**,**ValueColumn**,**SummaryColumn**,[**Function**])								
6		Parameters	**Data** - TableData or Array **ValueColumn** - Column Index of the value to be summarized. (Example: Sales) **SummaryColumn** - Column Index of the Group By column (Example: Segment) **Function** - Function number as in Subtotal() (By default 9 for Sum)								
7											
8		Home									
9											
10		**Example:**									
11											
12		**=BI_Summary(SalesTable,12,7)**									
13											
14		Consumer	1161401								
15		Corporate	706146.4								
16		Home Office	429653.1								
17											
18											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_TextSplitByPositions								
3		Category	Text								
4		Description	Returns an array of text split by number of positions as in Power Query. Useful for dynamically separate text to columns. Note: Unlike Power Query the column numbers are starting from 1 and not 0.								
5		Usage	=BI_TextSplitByPositions(**Text**,**Positions**)								
6		Parameters	**Text** - Text to be splitted by positions **Positions** - An array of column positions to split.								
7											
8		Home									
9											
10		**Example:**									
11											
12		--									
13		Date Doc Re Supplier Name Purchase Account Amount Company Verifi									
14		--									
15		12-12-2021 1 GRAHAM AGENCIES & INDUSTR CHEMICAL PURCHASES 16,00,000.00 RAJKUMAR									
16		14-11-2021 2 SRI KRISHNA CHEMICAL INDU CHEMICAL PURCHASES 3,100.00 RAJKUMAR									
17		14-11-2021 3 RAMAVILAS TUBE WORKS, SIV TUBE PURCHASE 8,149.00 RAJKUMAR									
18		26-09-2021 4 PUNITHA LAKSHIMI TUBE WOK TUBE PURCHASE 9,600.00 RAJKUMAR Verifi									
19											
20											
21		1	7	17	26	52	71	88	97	106	
22											
23				----------	---------	--------------------------	-------------------	-----------------	---------	---------	
24				Date	Doc Re	Supplier Name	Purchase Account	Amount	Company	Verifi	
25				----------	---------	--------------------------	-------------------	-----------------	---------	---------	
26				12-12-2021	1	GRAHAM AGENCIES & INDUSTR	CHEMICAL PURCHASES	16,00,000.00	RAJKUMAR		
27				14-11-2021	2	SRI KRISHNA CHEMICAL INDU	CHEMICAL PURCHASES	3,100.00	RAJKUMAR		
28				14-11-2021	3	RAMAVILAS TUBE WORKS, SIV	TUBE PURCHASE	8,149.00	RAJKUMAR		
29				26-09-2021	4	PUNITHA LAKSHIMI TUBE WOK	TUBE PURCHASE	9,600.00	RAJKUMAR	Verifi	
30											
31		**=BI_TextSplitByPositions(B12,$B$21#)**									
32											
33		**={1,7,17,26,52,71,88,97,106}**									
34											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_TopN								
3		Category	Statistics								
4		Description	Returns Top N ranked rows of a column of data.								
5		Usage	=BI_TopN(**DataRange**,**IndexColumn**,**N**)								
6		Parameters	**DataRange** - TableData or Array **IndexColumn** - Column Number to be classified **N** - Number of Top Records								
7											
8		Home									
9											
10		**Example:**									
11											
12		5		**=BI_TopN(CHOOSECOLS(SalesTable,1,2,6,12),4,B12)**							
13											
14		2698	CA-2014-145317	SM-20320	22638.48						
15		6827	CA-2016-118689	TC-20980	17499.95						
16		8154	CA-2017-140151	RB-19360	13999.96						
17		2624	CA-2017-127180	TA-21385	11199.97						
18		4191	CA-2017-166709	HL-15040	10499.97						
19											
20											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_TopPercent								
3		Category	Statistics								
4		Description	Returns Top N Percent of a column of Data.								
5		Usage	=BI_TopPercent(**Data**,**IndexColumn**, **Percent**)								
6		Parameters	**Data** - TableData or Array **IndexColumn** - Column Number to be classified **Percent** - Percentage of data to be extracted								
7											
8		Home									
9											
10		**Example:**									
11											
12		5%		**=BI_TopPercent(CHOOSECOLS(SalesTable,1,2,6,12),4,B12)**							
13											
14		2698	CA-2014-145317	SM-20320	22638.48						
15		6827	CA-2016-118689	TC-20980	17499.95						
16		8154	CA-2017-140151	RB-19360	13999.96						
17		2624	CA-2017-127180	TA-21385	11199.97						
18		4191	CA-2017-166709	HL-15040	10499.97						
19		9040	CA-2016-117121	AB-10105	9892.74						
20		4099	CA-2014-116904	SC-20095	9449.95						
21		4278	US-2016-107440	BS-11365	9099.93						
22		8489	CA-2016-158841	SE-20110	8749.95						
23		6426	CA-2016-143714	CC-12370	8399.976						
24											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Unpivot								
3		Category	Data								
4		Description	Returns Unpivoted Data like Power Query.								
5		Usage	=BI_Unpivot(**UnpivotData**,[**RemainingData**])								
6		Parameters	**UnpivotData** - Data to remove pivot **RemainingData** - Remaining Data in the table to be retained (By default all the columns are used for unpivoting)								
7											
8		Home									
9											
10		**Example:**									
11											
12	1	**When all the columns are unpivoted**									
13		April	May	June	July						
14		1500	1600	1550	1250						
15		2000	2500	1540	3245						
16		2500	2540	3200	1545						
17		2350	2650	3120	3650						
18		2850	2640	1890	3250						
19		4000	2740	3240	1240						
20											
21		Attributes	Values		**=BI_Unpivot(B13:E19)**						
22		April	1500								
23		May	1600								
24		June	1550								
25		July	1250								
26		April	2000								
27		May	2500								
28		June	1540								
29		July	3245								
30		April	2500								
31		May	2540								
32		June	3200								
33		July	1545								
34		April	2350								
35		May	2650								
36		June	3120								
37		July	3650								
38		April	2850								
39		May	2640								
40		June	1890								
41		July	3250								
42		April	4000								
43		May	2740								
44		June	3240								
45		July	1240								
46											

	A	B	C	D	E	F	G	H	I	J	K
47	2	**When some columns are remaining**									
48											
49											
50		RollNo	Name	Mark1	Mark2	Mark3					
51		1	A	45	65	65					
52		2	B	78	64	78					
53		3	C	65	34	48					
54		4	D	45	25	35					
55		5	E	18	48	45					
56											
57		RollNo	Name	Attributes	Values		**=BI_Unpivot(D50:F55,B50:C55)**				
58		1	A	Mark1	45						
59		1	A	Mark2	65						
60		1	A	Mark3	65						
61		2	B	Mark1	78						
62		2	B	Mark2	64						
63		2	B	Mark3	78						
64		3	C	Mark1	65						
65		3	C	Mark2	34						
66		3	C	Mark3	48						
67		4	D	Mark1	45						
68		4	D	Mark2	25						
69		4	D	Mark3	35						
70		5	E	Mark1	18						
71		5	E	Mark2	48						
72		5	E	Mark3	45						
73											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_UnpivotExceptFirst								
3		Category	Data								
4		Description	Returns Unpivoted Data leaving the first column as the attribute column.								
5		Usage	=BI_UnpivotExceptFirst(**Data**)								
6		Parameters	**Data** - TableData or Array including headers where the first column is retained and other columns are unpivoted.								
7											
8		Home									
9											
10		**Example:**									
11											
12			Consumer	Corporate	Home Office						
13		Accessories	87105.24	48190.56	32084.52						
14		Appliances	52819.58	36588.68	18123.9						
15		Art	14251.93	8590.448	4276.414						
16		Binders	118161	51560.31	33691.41						
17		Bookcases	68632.73	34005.92	12241.34						
18		Chairs	172862.7	99140.88	56445.48						
19		Copiers	69819.07	46829.39	32879.57						
20		Envelopes	7771.146	5942.67	2762.586						
21		Fasteners	1680.942	783.29	560.048						
22		Furnishings	49620.05	25001.27	17083.85						
23		Labels	6709.262	4101.646	1675.404						
24		Machines	79542.83	60276.76	49419.05						
25		Paper	36324.37	23883.13	18271.71						
26		Phones	169932.8	91153.41	68920.88						
27		Storage	100492.4	79791	43560.21						
28		Supplies	25741.5	19435.28	1496.758						
29		Tables	99933.8	70871.72	36160.02						
30											
31											
32			Attributes	Values		**=BI_UnpivotExceptFirst(B12#)**					
33		Accessories	Consumer	87105.24							
34		Accessories	Corporate	48190.56							
35		Accessories	Home Office	32084.52							
36		Appliances	Consumer	52819.58							
37		Appliances	Corporate	36588.68							
38		Appliances	Home Office	18123.9							
39		Art	Consumer	14251.93							
40		Art	Corporate	8590.448							
41		Art	Home Office	4276.414							
42		Binders	Consumer	118161							
43		Binders	Corporate	51560.31							
44		Binders	Home Office	33691.41							
45		Bookcases	Consumer	68632.73							
46		Bookcases	Corporate	34005.92							
47		Bookcases	Home Office	12241.34							
48		Chairs	Consumer	172862.7							
49		Chairs	Corporate	99140.88							
50		Chairs	Home Office	56445.48							
51		Copiers	Consumer	69819.07							
52		Copiers	Corporate	46829.39							

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_WDVSchedule								
3		Category	Financial								
4		Description	Returns Written Down Value (Declining Balance) Method of Depreciation Schedule.								
5		Usage	=BI_WDVSchedule(**Cost**,**Life**,**Scrap**)								
6		Parameters	**Cost** - Cost of the Asset **Life** - Life of the Asset **Scrap** - Scrap/Residual value at the end of the life.								
7											
8		Home									
9											
10		**Example:**									
11											
12		Cost	1000000		**=BI_WDVSchedule(C12,C13,C14)**						
13		Life	10								
14		Scrap	100000								
15											
16		WDV Rate :	0.205672								
17		Year	Op.Balance	Depreciation	Cl.Balance						
18		1	1000000	205671.8	794328.2						
19		2	794328.2	163370.9	630957.3						
20		3	630957.3	129770.1	501187.2						
21		4	501187.2	103080.1	398107.2						
22		5	398107.2	81879.4	316227.8						
23		6	316227.8	65039.12	251188.6						
24		7	251188.6	51662.41	199526.2						
25		8	199526.2	41036.91	158489.3						
26		9	158489.3	32596.78	125892.5						
27		10	125892.5	25892.54	100000						
28		Total		900000							
29											
30											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_WordCount								
3		Category	Text								
4		Description	Returns the number of words in text.								
5		Usage	=BI_WordCount(**Text**)								
6		Parameters	**Text** - Text used for word count. A range of text can also be used to count all the words								
7											
8		Home									
9											
10		**Example:**									
11											
12											
13		Effective paragraphs are the fundamental units of academic writing; consequently, the thoughtful, multifaceted arguments that your professors expect depend on them. Without good paragraphs, you simply cannot clearly convey sequential points and their relationships to one another.									
14											
15		37		**=BI_WordCount(B13)**							
16											
17		55607		**=BI_WordCount(SalesTable[Product Name])**							
18											
19											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_Workday.Indian								
3		Category	Indian								
4		Description	Returns the date after a specified number of Indian workdays (Second and Fourth Saturdays as holidays)								
5		Usage	=BI_Workday.Indian(**StartDate**,**Days**,**Holidays**)								
6		Parameters	**StartDate** - Starting Date of the Project **Days** - Number of Working Days **Holidays** - List of Holidays								
7											
8		Home									
9											
10		**Example:**									
11											
12		Start Date	15-10-2023								
13		No. of Working Days	100								
14											
15		Holidays									
16		12-11-2023									
17		25-12-2023									
18		31-12-2023									
19		01-01-2024									
20		14-01-2024									
21		26-01-2024									
22											
23		Day of completion	22-02-2024								
24											
25			**=BI_WorkDay.Indian(C12,C13,B16:B21)**								
26											
27											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_YearlyCalendar								
3		Category	Date								
4		Description	Returns the Yearly Calendar of any date or current date in horizontal or vertical format.								
5		Usage	=BI_YearlyCalendar([**Year**],[**Format**],[**Vertical**])								
6		Parameters	**Year** - Year number from 1900 to 9999 (By default current year) **Format** - Day Format - DDD - Short Form, DDDD - Long Form (By default short form) **Vertical** - True for vertical format of calendar (By default it is false)								
7											
8		Home									
9											
10		**Example:**	**=BI_YearlyCalendar()**								
11											
12		ஜனவிர	2023								
13		ஞோ ியுற	திங்ள	செ்வவேர்ய	◌ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ ின			
14		1	2	3	4	5	6	7			
15		8	9	10	11	12	13	14			
16		15	16	17	18	19	20	21			
17		22	23	24	25	26	27	28			
18		29	30	31							
19											
20		பிபரவிர	2023								
21		ஞோ ியுற	திங்ள	செ்வவேர்ய	◌ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ ின			
22					1	2	3	4			
23		5	6	7	8	9	10	11			
24		12	13	14	15	16	17	18			
25		19	20	21	22	23	24	25			
26		26	27	28							
27											
28		ே◌ர்ர்ச	2023								
29		ஞோ ியுற	திங்ள	செ்வவேர்ய	◌ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ ின			
30					1	2	3	4			
31		5	6	7	8	9	10	11			
32		12	13	14	15	16	17	18			
33		19	20	21	22	23	24	25			
34		26	27	28	29	30	31				
35											
36		ஏ்பர்ல	2023								
37		ஞோ ியுற	திங்ள	செ்வவேர்ய	◌ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ ின			
38								1			
39		2	3	4	5	6	7	8			
40		9	10	11	12	13	14	15			
41		16	17	18	19	20	21	22			
42		23	24	25	26	27	28	29			
43		30									
44											
45		மே	2023								
46		ஞோ ியுற	திங்ள	செ்வவேர்ய	◌ுபத்ன	வியோழ்ன	சவ்ளிள	ெ◌ ின			
47			1	2	3	4	5	6			
48		7	8	9	10	11	12	13			
49		14	15	16	17	18	19	20			
50		21	22	23	24	25	26	27			
51		28	29	30	31						
52											

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Function Name	BI_ZScore								
3		Category	Statistics								
4		Description	Returns the Z Score of data.								
5		Usage	=BI_Zscore(**Data**,[**Population**])								
6		Parameters	**Data** - TableData or Array **Population** - False if the data is a sample. (By default false)								
7											
8		Home									
9											
10		**Example:**									
11											
12		Benford Law First Two Digits									
13		Digit	Frequency	Actual	Expected						
14		10	411	0.047614	0.041393	3.599557		**=BI_ZScore(D14:D103,TRUE)**			
15		11	387	0.044833	0.037789	3.325382					
16		12	377	0.043675	0.034762	3.211143					
17		13	260	0.03012	0.032185	1.874542					
18		14	329	0.038114	0.029963	2.662794					
19		15	305	0.035334	0.028029	2.388619					
20		16	222	0.025718	0.026329	1.440432					
21		17	241	0.027919	0.024824	1.657487					
22		18	210	0.024328	0.023481	1.303345					
23		19	269	0.031163	0.022276	1.977357					
24		20	207	0.023981	0.021189	1.269073					
25		21	166	0.019231	0.020203	0.800691					
26		22	156	0.018072	0.019305	0.686452					
27		23	191	0.022127	0.018483	1.08629					
28		24	160	0.018536	0.017729	0.732148					
29		25	202	0.023401	0.017033	1.211953					
30		26	117	0.013554	0.01639	0.240918					
31		27	160	0.018536	0.015794	0.732148					
32		28	105	0.012164	0.01524	0.103831					
33		29	132	0.015292	0.014723	0.412277					
34		30	132	0.015292	0.01424	0.412277					
35		31	142	0.01645	0.013788	0.526517					
36		32	120	0.013902	0.013364	0.27519					
37		33	112	0.012975	0.012965	0.183799					
38		34	105	0.012164	0.012589	0.103831					
39		35	121	0.014018	0.012234	0.286614					
40		36	111	0.012859	0.011899	0.172375					
41		37	91	0.010542	0.011582	-0.0561					
42		38	97	0.011237	0.011281	0.012439					
43		39	106	0.01228	0.010995	0.115255					
44		40	89	0.01031	0.010724	-0.07895					
45		41	94	0.01089	0.010465	-0.02183					
46		42	46	0.005329	0.010219	-0.57018					
47		43	94	0.01089	0.009984	-0.02183					
48		44	83	0.009615	0.00976	-0.1475					
49		45	94	0.01089	0.009545	-0.02183					