Finance Automation Portfolio – Excel LAMBDA Functions & Power Query

By Vishnukumar Ganesan

This portfolio demonstrates 13+ custom LAMBDA functions and Power Query automations for financial reporting, reconciliations, and data cleaning.

S.No	Function Name	Use Case (Detailed)	Key Excel Concepts Used
1	VK_RemoveDuplicates	Stacks up to 4 arrays using VSTACK(), removes duplicates with UNIQUE(), and filters out empty cells dynamically.	UNIQUE, VSTACK, FILTER
2	VK_Extract_Text	Extracts a specific number of characters from a text cell starting from the position of a search string.	MID, SEARCH
3	VK_Vlookup_Automation	Automates VLOOKUP with dynamic column selection using MATCH() and returns 0 if no match is found.	VLOOKUP, MATCH, IFERROR
4	VK_Dual_Vlookup	Performs a VLOOKUP using two combined lookup values for stricter matching.	VLOOKUP, MATCH, IFERROR
5	VK_XLookup_Automation	Enables a two-way lookup (row and column) using nested XLOOKUP functions for dynamic table queries.	XLOOKUP
6	VK_XLOOKUP_Multiple	Performs multi-condition lookups dynamically using XLOOKUP() with ISOMITTED() to handle optional parameters.	XLOOKUP, ISOMITTED
7	VK_Check_Availability	Checks if a specific value exists within an array and returns “Available” or “Not Available”.	MATCH, ISNUMBER
8	VK_Net_Amount	Returns the debit or credit amount dynamically based on a Dr or Cr indicator input.	IF
9	VK_Number_Extract	Extracts the first numeric character in a text string and returns all subsequent characters.	MID, SEARCH
10	VK_Min_Value_Lookup	Finds the value corresponding to the **minimum value** in a dataset filtered by a lookup condition.	XLOOKUP, MINIFS
11	VK_Max_Value_Lookup	Finds the value corresponding to the **maximum value** in a dataset filtered by a lookup condition.	XLOOKUP, MAXIFS
12	VK_Next_Min_Lookup	Finds the next smallest (2nd, 3rd, etc.) value and returns the corresponding result dynamically.	XLOOKUP, SMALL
13	VK_Next_Max_Lookup	Finds the next largest (2nd, 3rd, etc.) value and returns the corresponding result dynamically.	XLOOKUP, LARGE

VK_RemoveDuplicates

Purpose: Stacks up to 4 arrays using VSTACK(), removes duplicates with UNIQUE(), and filters out empty cells dynamically.

#NAME?	=LAMBDA(Array1,Array2,Array3,Array4,
	  UNIQUE(
	  FILTER(
	  VSTACK(
	  Array1,
	  IF(ISOMITTED(Array2),"",Array2),
	  IF(ISOMITTED(Array3),"",Array3),
	  IF(ISOMITTED(Array4),"",Array4)
	  ),
	  VSTACK(
	  Array1,
	  IF(ISOMITTED(Array2),"",Array2),
	  IF(ISOMITTED(Array3),"",Array3),
	  IF(ISOMITTED(Array4),"",Array4)
	  )<>""
	  )
	  )
	)

Array1	Array2	Array3	Array4	Output
A	B	C		A, B, C
B	D	F		A, B, C, D, F

VK_Extract_Text

Purpose: Extracts a specific number of characters starting from the position of a search text.

#NAME?	=LAMBDA(Cell_Ref,Text_to_Search,No_of_Digit_req, MID(Cell_Ref, SEARCH(Text_to_Search, Cell_Ref), No_of_Digit_req))

Cell	Text_to_Search	No_of_Digit_req	Output
Order#12345XYZ	#	6	#12345
Invoice-9876AB	-	5	-9876

VK_Vlookup_Automation

Vlookup_Automation automates VLOOKUP with dynamic column selection, reducing manual effort and preventing errors.

#NAME?	=LAMBDA(Lookup_Value1, Lookup_Array, Column_Value, Column_Range,
	  IFERROR(
	  VLOOKUP(
	  Lookup_Value1,
	  Lookup_Array,
	  MATCH(Column_Value, Column_Range, 0),
	  0
	  ),
	  0
	  )
	)

Name	Age	City
Alex	25	London
Ben	30	Paris

Lookup_Value1 = Ben

Column_Value = City

Output: Paris

VK_Dual_Vlookup

This DualLookup function performs a VLOOKUP with exactly 2 lookup values, ensuring a strict requirement for both inputs

#NAME?	=LAMBDA(Lookup_Value1, Lookup_Value2, Lookup_Array, Column_Value, Column_Range, IFERROR(VLOOKUP(Lookup_Value1 & Lookup_Value2, Lookup_Array, MATCH(Column_Value, Column_Range, 0), 0), 0))

Product	Region	Price
Pen	East	10
Pen	West	12

Lookup: Pen + West → **Output:** 12

VK_XLookup_Automation

XLookup_Automation performs a dynamic two-way lookup using XLOOKUP, finding the correct column first and then retrieving the row value.

#NAME?	=LAMBDA(Row_Lookup, Row_Array, Return_Array, Col_Lookup, Col_Array, XLOOKUP(Row_Lookup, Row_Array, XLOOKUP(Col_Lookup, Col_Array, Return_Array, "Column Heading Not Found", 0), "Not Found Data", 0))

	Jan	Feb
A	10	12
B	20	22

Row_Lookup = B, Col_Lookup = Feb → Output: 22

VK_XLOOKUP_Multiple

VK_XLOOKUP_Advanced performs a dual-condition lookup with optional criteria, dynamically filtering rows and columns to return matching data with error handling.

#NAME?	=LAMBDA(Lookup_Value1, Lookup_Value2_Optional, Lookup_Range1, Lookup_Range2_Optional, Column_Value1, Column_Value2_Optional, Column_Range1, Column_Range2_Optional, Data_Range, XLOOKUP(1, (Lookup_Range1 = Lookup_Value1) * (IF(ISOMITTED(Lookup_Value2_Optional), 1, (Lookup_Range2_Optional = Lookup_Value2_Optional))), XLOOKUP(1, ((Column_Range1 = Column_Value1) * IF(ISOMITTED(Column_Value2_Optional), 1, (Column_Range2_Optional = Column_Value2_Optional))), Data_Range, "Column Header Not Found", 0), "Data Not Found")

Name	Dept	Jan	Feb	Mar
Alex	HR	1000	1200	1400
Alex	IT	1100	1250	1450
Ben	HR	900	950	1000
Ben	IT	850	900	950

VK_Check_Availability

VK_Check_Availability verifies if a value exists in a specified range and returns "Available" or "Not Available" accordingly.

#NAME?	=LAMBDA(Lookup_Value, Lookup_Array, IF(ISNUMBER(MATCH(Lookup_Value, Lookup_Array, 0)), "Available", "Not Available"))

A (Lookup Value)	B (Lookup Array)	Result (C)
Apple	Apple	Available
Mango	Orange	Not Available
Orange	Grape	Available
Banana	Apple	Not Available
Grape	Orange	Available

VK_Net_Amount

VK_Net_Amount returns the Debit_Amount when DrCR is "D", otherwise returns the Credit_Amount.

#NAME?	=LAMBDA(DrCR, Debit_Amount, Credit_Amount, IF(DrCR = "D", Debit_Amount, Credit_Amount))

A (DrCR)	B (Debit Amount)	C (Credit Amount)	Result (D)
D	1000	0	1000
C	200	500	500
D	300	50	300
C	0	700	700
D	150	40	150

VK_Number_Extract

#NAME?	=LAMBDA(Cell_Ref, MID(Cell_Ref, MIN(SEARCH({0,1,2,3,4,5,6,7,8,9}, Cell_Ref & "0123456789")), LEN(Cell_Ref) - MIN(SEARCH({0,1,2,3,4,5,6,7,8,9}, Cell_Ref & "0123456789")) + 1))

A (Input Text)	Output (Extracted)
INV12345XYZ	12345XYZ
AB45CD6	45CD6
NO123	123
TEST5000	5000

XYZ	#VALUE! (no number)

VK_Min_Value_Lookup

#NAME?	=LAMBDA(Lookup_Value, Lookup_Range, Min_Value_Range, Return_Range, XLOOKUP(1, (Lookup_Range = Lookup_Value) * (Min_Value_Range = MINIFS(Min_Value_Range, Lookup_Range, Lookup_Value)), Return_Range, 0, 0))

A (Product)	B (Price)	C (Supplier)
Apple	50	Supplier A
Apple	40	Supplier B
Apple	60	Supplier C
Mango	70	Supplier D
Mango	65	Supplier E

VK_Max_Value_Lookup

#NAME?	=LAMBDA(Lookup_Value, Lookup_Range, Max_Value_Range, Return_Range, XLOOKUP(1, (Lookup_Range = Lookup_Value) * (Max_Value_Range = MAXIFS(Max_Value_Range, Lookup_Range, Lookup_Value)), Return_Range, 0, 0))

A (Product)	B (Price)	C (Supplier)
Apple	50	Supplier A
Apple	40	Supplier B
Apple	60	Supplier C
Mango	70	Supplier D
Mango	65	Supplier E

VK_Next_Min_Lookup

#NAME?	=LAMBDA(Lookup_Value, MinValue_Num, Lookup_Range, MinValue_Lookup_Range, Return_Range, XLOOKUP(1, (Lookup_Range = Lookup_Value) * (MinValue_Lookup_Range = SMALL(MinValue_Lookup_Range, MinValue_Num)), Return_Range, 0, 0))

A (Product)	B (Price)	C (Supplier)
Apple	50	Supplier A
Apple	40	Supplier B
Apple	60	Supplier C
Apple	45	Supplier D
Mango	70	Supplier E

VK_Next_Max_Lookup

#NAME?	=LAMBDA(Lookup_Value, MaxValue_Num, Lookup_Range, MaxValue_Lookup_Range, Return_Range, XLOOKUP(1, (Lookup_Range = Lookup_Value) * (MaxValue_Lookup_Range = LARGE(MaxValue_Lookup_Range, MaxValue_Num)), Return_Range, 0, 0))

A (Product)	B (Price)	C (Supplier)
Apple	50	Supplier A
Apple	40	Supplier B
Apple	60	Supplier C
Apple	45	Supplier D
Mango	70	Supplier E