PGDM Finance

PGDM Finance students aiming for roles at Deloitte need to master specific Excel skills that are essential for finance, accounting, and business analysis tasks. Key Excel skills include:

- **Advanced Lookup Functions:** VLOOKUP, HLOOKUP, and the powerful INDEX-MATCH combination for data retrieval from complex tables.
- **Financial Formulas:** NPV (Net Present Value), IRR (Internal Rate of Return), PMT (Payment function) used for financial modeling and investment analysis.
- **Pivot Tables:** Crucial for summarizing, analyzing, and presenting large datasets dynamically and efficiently.
- **Data Visualization and Charting:** To create insightful, clear financial reports for stakeholders.
- **What-If Analysis:** Tools like Scenario Manager and Goal Seek for forecasting and decision-making.
- **Macros and VBA:** Automate repetitive tasks and increase efficiency for complex financial modeling.
- **Concatenation and Text Functions:** For joining and cleaning data in reports.
- **Spreadsheet Formatting**: To organize and present data clearly and professionally.
- **Data Validation and Management:** Ensuring data accuracy and integrity in financial models.

These skills combined help in building robust financial models, performing data analysis, creating reports, and supporting decision-making processes vital for Deloitte finance roles.

The most frequently used Excel functions in Deloitte finance roles include:

- **Lookup Functions:**
  - VLOOKUP, HLOOKUP, and especially the more flexible INDEX and MATCH combination for retrieving data from tables
  - XLOOKUP for powerful and exact data searching in any direction
- **Financial Functions:**
  - NPV (Net Present Value) and XNPV for calculating present values of cash flows

    - IRR and XIRR for determining internal rates of return with periodic and non-periodic dates
    - PMT for calculating loan payments
    - IPMT for interest portion of payments
    - RATE for yield calculations on bonds or loans
- **Data Manipulation and Analysis:**
    - OFFSET to reference cells at offsets from a starting point
    - CHOOSE for selecting values from a list to create scenarios
    - IF for conditional logic
    - SUMIF for summing with conditions
- **Date and Time Functions:**
    - TODAY to get current date
    - EOMONTH for calculating month-end dates
- **Other useful functions seen frequently:**
    - DSUM, DMAX, DAVERAGE for database-like calculations on filtered data

These functions support tasks like financial modeling, valuation, budgeting, reporting, and scenario analysis which are central to Deloitte finance roles.fe+3

Standard Aggregating Functions

Standard aggregating functions are fundamental to almost all Excel work and are used to add up a range of numbers and find the total, minimum, maximum, average and count.

- SUM
- MIN
- MAX
- AVERAGE
- COUNT

These functions all take a range or multiple cells/values as their arguments. Using the SUM function as an example, the syntax is:

=SUM(value1,value2)

or

=SUM(range1, range2)

Note: *The syntax is the same for the other functions listed above.*

While these are some of the easiest functions in Excel's library, there are some lesser known tricks for the SUM function you might want to check out here:

Conditional Aggregating Functions

Conditional aggregating functions enable you to specify criteria that must be met before a value is included in the aggregation.

They include:

- SUMIFS
- MINIFS
- MAXIFS
- AVERAGEIFS
- COUNTIFS

These functions all take a range, criteria range and criteria as their arguments.Using the SUMIFS function as an example, the syntax is:

=SUMIFS(sum_range, criteria_range1, criteria1,...)

Note: *The syntax is the same for the other functions listed above.*

Logical Functions

Logical functions enable you to write 'if' statements where if one logical test is true, a calculation or value is returned, otherwise a different calculation or result is returned.

They are handy for modeling different scenarios and outcomes based on a range of assumptions.

The functions and their syntax in this section are:

IF Function

=IF(logical_test, value_if_true, value_if_false)

Nested IF Functions

=IF(logical_test, value_if_true,IF(logical_test, value_if_true, value_if_false))

IFS Function

=IFS(logical_test,value_if_true,logical_test2,value_if_true2...)

IF with AND Functions

=IF(AND(logical_test1,logical_test2…), value_if_true, value_if_false)

IF with OR Functions

=IF(OR(logical_test1,logical_test2…), value_if_true, value_if_false)

If you get stuck, use our IF formula builder and have your IF formulas written for you.

Lookup Functions

Lookup functions are used to look up a value in another table and return a corresponding value from the same row.

They can be used to create a reference table for financial modeling or bring data from one table into another.

For those with Excel 2021 or later, XLOOKUP should be your go-to lookup function. It overcomes the limitations of VLOOKUP and can do everything INDEX & MATCH can do.

Syntax:

=XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])

If you have an earlier version of Excel, I encourage you to use INDEX & MATCH as an alternative to XLOOKUP.

Financial Functions

There are a ton of financial functions in Excel which make easy work of calculating common financial metrics.

The financial functions listed below are some of the functions you'll most commonly use as a financial modeler:

- PMT – Periodic Payment
- PV – Present Value
- NPV – Net Present Value
- IRR – Internal Rate of Return

Check out the video above for step-by-step examples of these functions.

PMT Function

The PMT function is used to calculate the periodic payment for a loan or investment. It can be used to model debt repayments or investment returns in a financial model.

Syntax:

PMT(rate, nper, pv, [fv], [type])

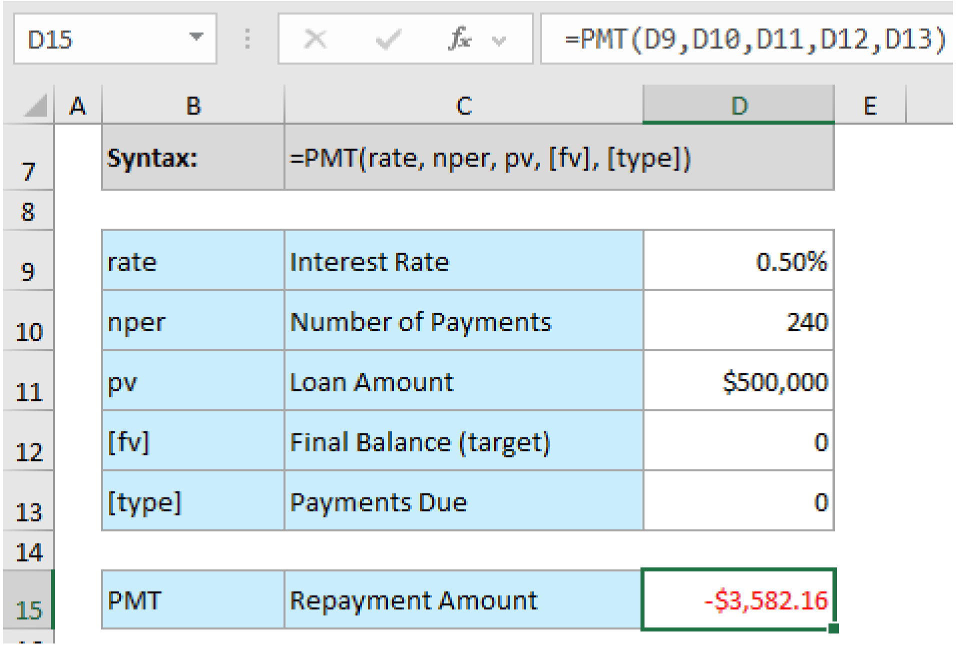
The ***rate*** argument is the rate per payment period.

The ***nper*** argument is the number of payments made over the length of the loan. e.g. for a 20-year loan with monthly repayments there would be 240 payments

The ***pv*** argument is the principal or loan amount

The ***fv*** argument is optional. It represents the final balance of the loan or target. If omitted, zero is assumed.

The ***type*** argument is optional. It represents when payments are due. 0 = end of period , 1 = beginning of period. If omitted, zero is assumed.



D15 =PMT(D9,D10,D11,D12,D13)

	A	B	C	D	E
7		**Syntax:**	=PMT(rate, nper, pv, [fv], [type])		
8					
9		rate	Interest Rate	0.50%	
10		nper	Number of Payments	240	
11		pv	Loan Amount	$500,000	
12		[fv]	Final Balance (target)	0	
13		[type]	Payments Due	0	
14					
15		PMT	Repayment Amount	-$3,582.16	

*Note: results returned may differ from your bank, as there may be additional fees and taxes.

They may also calculate interest daily, or you may have an offset account etc.

PV Function

The PV function calculates the present value of a future cash flow.

Syntax:

PV(rate, nper, pmt, [fv], [type])

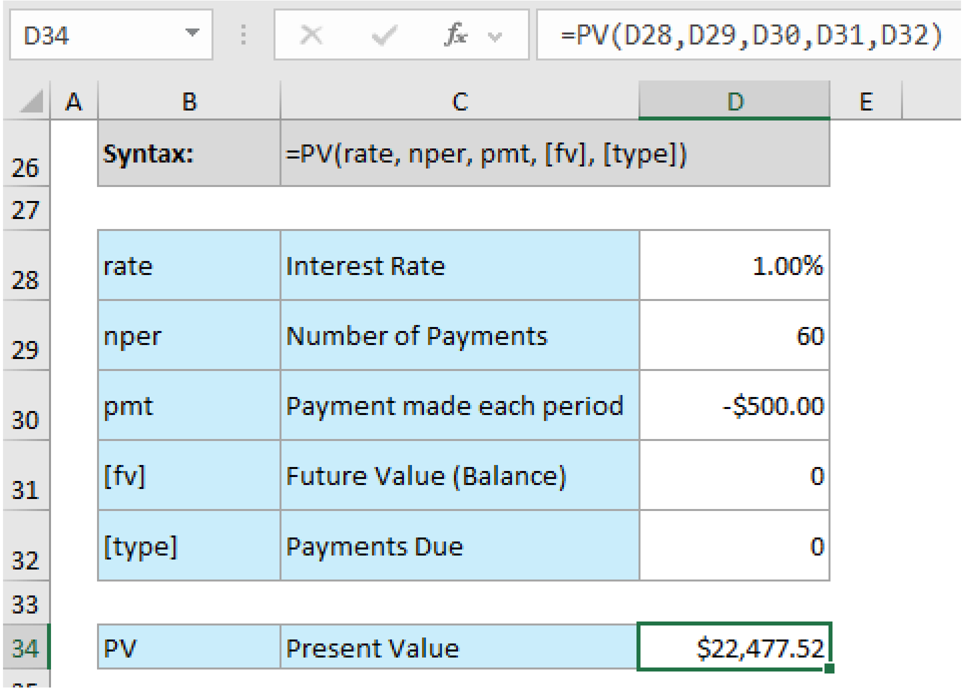
The ***rate*** argument is the rate per payment period.

The ***nper*** argument is the number of payments made over the length of the loan. e.g. for a 5-year loan with monthly repayments there would be 60 payments

The ***pmt*** argument represents the payment made each period and cannot change over the life of the annuity.

The ***fv*** argument is optional. It represents the future value, or a cash balance you want to attain after the last payment is made.

The ***type*** argument is optional. 0 = end of period , 1 = beginning of period. If omitted, zero is assumed.



=PV(D28,D29,D30,D31,D32)

Syntax:	=PV(rate, nper, pmt, [fv], [type])	
rate	Interest Rate	1.00%
nper	Number of Payments	60
pmt	Payment made each period	-$500.00
[fv]	Future Value (Balance)	0
[type]	Payments Due	0
PV	Present Value	$22,477.52

NPV Function

The NPV function is used to calculate the net present value of an investment based on a series of future cash flows.

It is a key function for discounted cash flow (DCF) analysis.

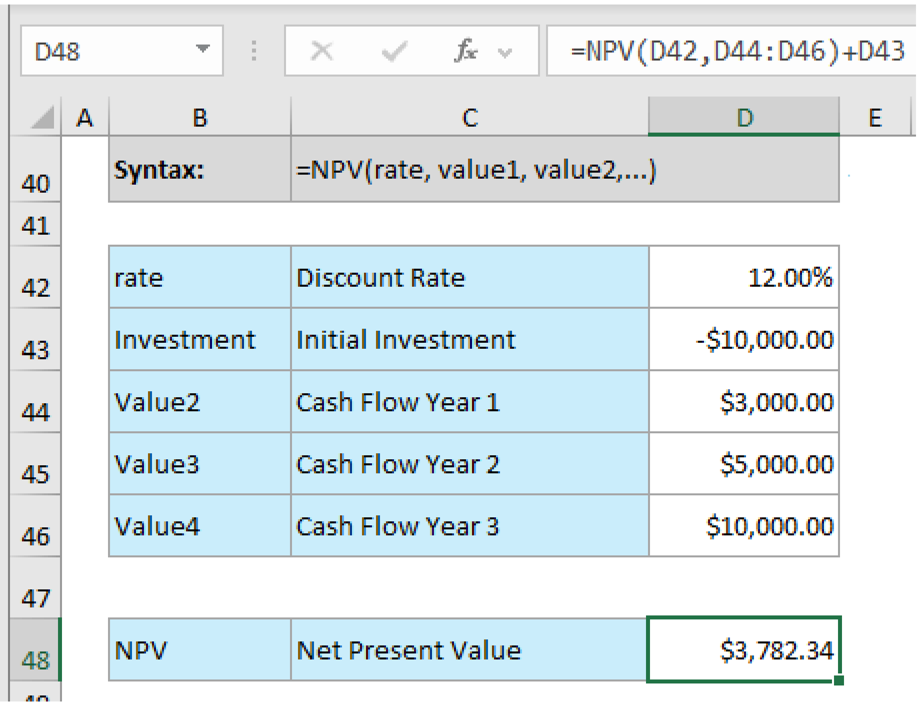
NPV assumes cash flows occur at the end of each period and requires them to be equally spaced. If not, use XNPV function.

Syntax: NPV(rate, value1, value2,...)

The ***rate*** argument is the rate of discount over the length of one period.

The ***value*** arguments represent the cash flows. Excel uses the order of value1, value2, ... to interpret the order of cash flows. Empty cells are ignored.

Note: if the initial investment occurs at the start of the investment period, do not include it in the vales as this should not be discounted. Instead, add it on outside the formula, as shown below.



D48 =NPV(D42,D44:D46)+D43

	A	B	C	D	E
40		**Syntax:**	=NPV(rate, value1, value2,...)		
41					
42		rate	Discount Rate	12.00%	
43		Investment	Initial Investment	-$10,000.00	
44		Value2	Cash Flow Year 1	$3,000.00	
45		Value3	Cash Flow Year 2	$5,000.00	
46		Value4	Cash Flow Year 3	$10,000.00	
47					
48		NPV	Net Present Value	$3,782.34	

IRR Function

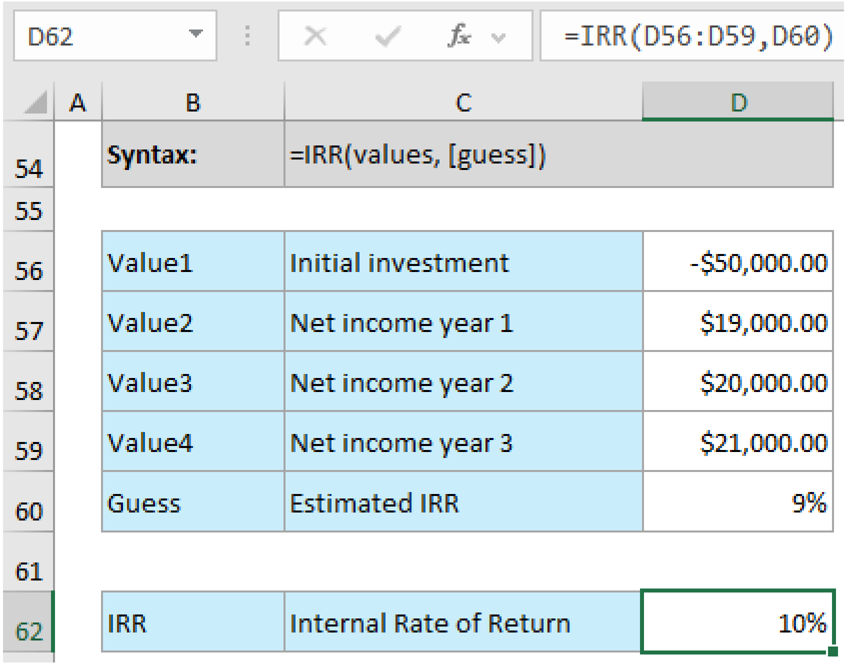
The IRR function is used to calculate the internal rate of return of an investment. It is another key function for DCF analysis.

Syntax: IRR(values, [guess])

Values is an array or a reference to cells that contain numbers for which you want to calculate the internal rate of return. They must contain at least one positive and one negative value.

IRR uses the order of values to interpret the order of cash flows. Therefore, you need to enter your payment and income values in the sequence you want.

The ***Guess*** argument is optional. It's a number that you guess is close to the result of IRR. If omitted, it's assumed to be 10%.



D62

=IRR(D56:D59,D60)

	A	B	C	D
54		**Syntax:**	=IRR(values, [guess])	
55				
56		Value1	Initial investment	-$50,000.00
57		Value2	Net income year 1	$19,000.00
58		Value3	Net income year 2	$20,000.00
59		Value4	Net income year 3	$21,000.00
60		Guess	Estimated IRR	9%
61				
62		IRR	Internal Rate of Return	10%

Date Functions

Nearly all financial models are based on data over time. Below are few common date functions you're likely to need:

- EOMONTH(start_date, months) – returns the last day of the month before or after a start date specified with a date serial number.
- EDATE(start_date, months) – rolls a date serial number forward or back based on the number of months specified in the 'months' argument.
- INTL(start_date, end_date, [weekend], [holidays]) - returns the number of whole workdays between two dates using parameters to indicate which and how many days are weekend days. Weekend days and any days that are specified as holidays are not considered as workdays.

How are lookup functions like VLOOKUP and INDEX-MATCH used in Deloitte roles?

In **Deloitte finance roles**, lookup functions like VLOOKUP and the combination of INDEX-MATCH are used extensively for retrieving and cross-referencing data efficiently in large datasets:

- **VLOOKUP** is commonly used to search for a value in the first column of a table and return a corresponding value from a specified column in the same row. It's straightforward and effective when the lookup column is the leftmost column in the data set and the structure is stable. For example, it can be used to fetch financial metrics, client details, or transaction data based on a unique identifier.
- **INDEX-MATCH** is favored for its greater flexibility and reliability. Unlike VLOOKUP, INDEX-MATCH:

  - Allows lookup values to be in any column (not necessarily the leftmost) and can return data from any column dynamically.
  - Is immune to errors caused by inserting or deleting columns, as it doesn't rely on static column index numbers.
  - Is better for complex datasets where columns may shift or when performing "right to left" lookups (looking up a value to the left of the key).
  - Handles large data sets more efficiently, often with faster calculation times.
  - Supports dynamic column referencing, making it easier to copy or drag formulas across cells without breakage.

In Deloitte projects, these functions help in financial analysis, valuation, budgeting, client reporting, and audit tasks by enabling analysts to quickly find and consolidate data from multiple tables or datasets accurately and flexibly. The INDEX-MATCH method is often preferred when the dataset is large or complex, while VLOOKUP may be used for simpler, fixed-table lookups.

What are the advantages of using INDEX-MATCH over VLOOKUP in Deloitte data analysis tasks?

The advantages of using INDEX-MATCH over VLOOKUP in Deloitte data analysis tasks include:

- **Lookup in Any Direction:** Unlike VLOOKUP, which can only search in the leftmost column and return data to the right, INDEX-MATCH can perform lookups both left-to-right and right-to-left, providing much greater flexibility.
- **Resilience to Column Changes:** VLOOKUP relies on a static column index number. If columns are inserted or deleted in the data table, VLOOKUP results can break or return incorrect values. INDEX-MATCH uses dynamic column references, so it remains accurate even when the table structure changes.
- **Fewer Errors:** Because INDEX-MATCH allows direct referencing of the return column range, it reduces errors related to incorrect column index numbers used in VLOOKUP.
- **No Lookup Value Length Limit:** VLOOKUP has a 255-character limit on the lookup value, whereas INDEX-MATCH has no such limit, making it more robust for long strings or complex keys.
- **Faster Performance on Large Data:** INDEX-MATCH only processes the necessary lookup and return columns instead of scanning the entire table array like VLOOKUP, which can make INDEX-MATCH faster in large datasets.

- **Easier to Copy and Drag Formulas:** INDEX-MATCH formulas can be copied or dragged more easily across cells without breaking, as it supports dynamic column referencing, unlike VLOOKUP that requires fixed column numbers.

These advantages make INDEX-MATCH a more flexible, robust, and efficient choice for handling complex and large-scale data analysis tasks typical in Deloitte finance roles.