9

Interactive Dashboards and Advanced Data Visualization

In this Chapter

Welcome to Chapter 9 of "Excel Data Analysis for the Modern Business"! In this chapter, we're going to dive into the exciting world of preparing

dashboards in Excel. Dashboards are like the superheroes of data visualization, helping us understand complex information at a glance. So buckle up, because we're about to embark on a journey that will transform your data into captivating and informative visual displays!

Interactive dashboards and advanced data visualization in

Dashboards provide a consolidated view of key metrics and insights, allowing users to interact with the data and gain real-time insights. Advanced data visualization techniques, such as heatmaps, small multiples, and infographics, go beyond traditional charts and graphs to convey complex information in a visually compelling manner.

9.1. Introduction to Dashboards:

Welcome to the captivating world of dashboards! In this section, we'll embark on a journey that will revolutionize the way you analyze and understand your business data. Dashboards serve as a game-changing tool for managers, offering a high-level overview and empowering quick decision-making. With the help of Excel, we'll explore the art of creating dynamic dashboards that provide analysis, insights, and timely alerts.

9.1.1. What is a Dashboard?

Dashboards are the superheroes of data visualization, allowing managers to grasp the pulse of their business at a glance. These versatile tools, also known as management dashboards, information dashboards, or dashboard reports, provide a concise yet comprehensive overview of key metrics. By condensing

complex information into easily digestible visual representations, dashboards assist managers in making informed decisions swiftly.

9.1.2. Origins and Evolution of Dashboards

To truly appreciate the significance of dashboards, let's take a step back and explore their roots. In the realm of information technology, a dashboard is an easy-to-read, real-time user interface that presents a graphical snapshot of an organization's or department's key performance indicators (KPIs). Drawing inspiration from the term "dashboard" in automobiles, where vital information is summarized for ease of use, business dashboards provide a similar functionality in the context of monitoring organizational performance.

Figure 9.1: Vehicle dashboard

The concept of digital dashboards emerged in the 1970s, but it was in the 1980s that they began to take shape in the business world. However, due to challenges related to data refreshing and handling, their development was temporarily halted. It wasn't until the 1990s, with the rapid advancement of the information age and the emergence of data warehousing, that dashboards experienced a

resurgence. Visionaries like David P. Norton recognized their potential as an integral part of decision-making processes.

9.1.3. Key Metrics and Benefits of Dashboards

Dashboards focus on presenting key metrics that provide valuable insights into the performance of an organization or department. By showcasing these metrics in a visually appealing and easily understandable format, dashboards enable managers to assess the current status and historical trends at a glance. The benefits of using dashboards are numerous, including the ability to track progress, identify patterns, and make data-driven decisions swiftly. Furthermore, dashboards promote transparency and collaboration within the organization, fostering a culture of informed decision-making.

9.1.4. Types of Dashboards and Live Data Updates

Dashboards come in various types, tailored to meet specific needs and objectives. Whether it's a strategic dashboard that provides an overview of the organization's long-term goals or an operational dashboard that monitors day-to-day activities, there's a dashboard type for every purpose. Additionally, we'll explore the concept of live data updates, which ensures that dashboards always reflect the most up-to-date information. This real-time aspect allows managers to stay on top of changes and respond promptly.

9.1.5. Excel as Your Dashboard Creation Tool

When it comes to creating and distributing dashboards, Microsoft Excel shines as a powerful and accessible tool. Its user-friendly interface and widespread availability make it an excellent choice for beginners and experienced professionals alike. In the following sections, we'll dive into the practical steps of crafting impressive dashboards using the capabilities of Excel.

In the next subsection, we'll explore the essential planning phase for creating dashboards. So get ready to transform your data into visually captivating insights as we embark on this exciting journey together.

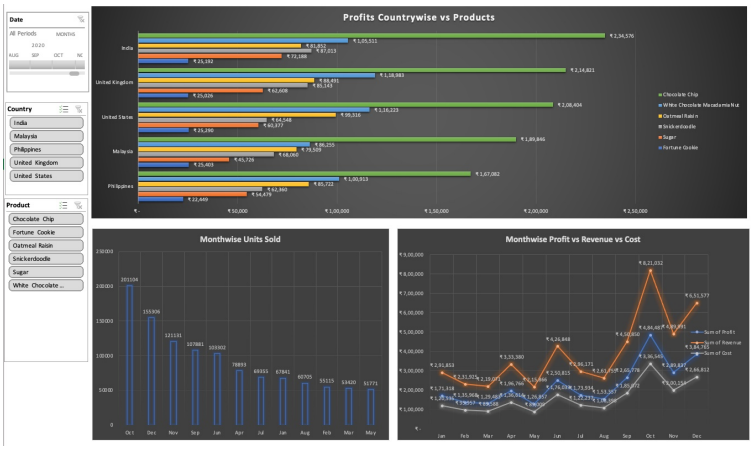


Figure 9.2: sales dashboard

9.2. Types of Dashboards:

In the exciting world of dashboards, different types emerge to cater to specific needs and objectives.**Let's explore the four primary types of dashboards:**

Infographics

Infographics are visual representations of information, data, or knowledge that present complex concepts in a concise and engaging format. They combine images, charts, diagrams, and text to tell a story or convey a message effectively. Infographics are designed to capture attention, simplify complex information, and make it more memorable and accessible to the audience.

1. Strategic Dashboards: Illuminating the Big Picture

Strategic dashboards are the go-to choice for managers at all levels of an organization when it comes to making critical decisions. These dashboards provide a snapshot of data, displaying the health and opportunities of the business. With a focus on high-level performance measures and forecasts, strategic dashboards offer a broad overview without delving into intricate details. Their purpose is to guide strategic planning and facilitate discussions around key initiatives. By capturing the essence of the organization's goals and progress, strategic dashboards empower managers to steer the ship towards success.

Figure 9.3: strategic dashboard

2. Analytical Dashboards: Diving Deep into Data

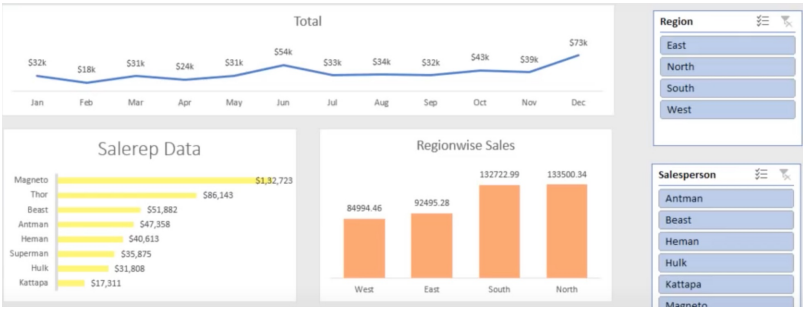


Figure 9.4: Analytics dashboard

Analytical dashboards are all about diving deep into the wealth of data to uncover insights and patterns. They provide more context, comparisons, and historical

perspectives, allowing for a comprehensive analysis of various facets of the business. Interactivity takes center stage in analytical dashboards, enabling users to drill down into underlying details and explore the data from multiple angles. Examples of analytical dashboards include Finance Management dashboards and Sales Management dashboards, which empower users to gain a comprehensive understanding of financial trends and sales performance.

3. Operational Dashboards: Monitoring Real-Time Performance

Operational dashboards are the go-to tools for monitoring real-time performance and ensuring smooth day-to-day operations. They focus on providing up-to-the-minute data on key operational metrics, allowing managers to make quick decisions and take necessary actions. These dashboards are often designed with simplicity and clarity in mind, offering a concise overview of performance indicators specific to operational areas. Whether it's tracking customer service response times or monitoring production line efficiency, operational dashboards provide real-time visibility to ensure everything runs smoothly.

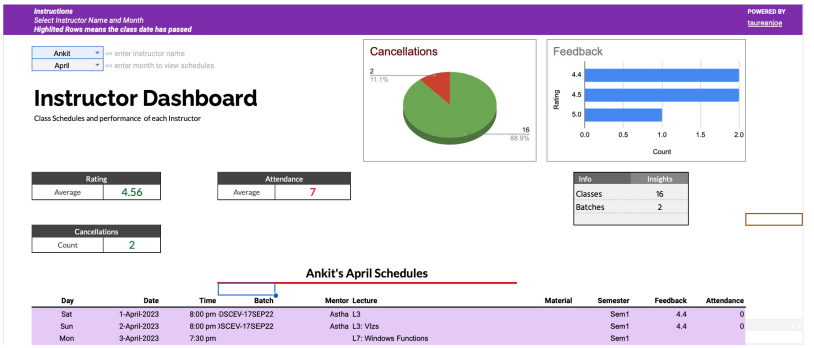


Figure 9.5: operational dashboard

4. Informational Dashboards: Sharing Insights with Stakeholders

Informational dashboards serve as a means of sharing insights and data-driven information with stakeholders, such as executives, clients, or team members. These dashboards are designed to communicate key metrics and progress towards

goals in a visually appealing and accessible manner. They are often static or periodically updated, providing a snapshot of data at specified intervals. While they may not have extensive interactivity, they still allow stakeholders to gain a clear understanding of the organization's performance and achievements.

Index Ticker	Live	Day Change %	Day Change	Last Close	30 Day Chart		
NIFTY_50	18,297.00	-0.10%	-18.10	18,315.10			
SENSEX	61,904.52	-0.06%	-35.68	61,940.20			
INDIA_VIX	13.22	1.07%	0.14	13.08			
Stock Ticker	**LTP**	**Day Change %**	**Day Change**	**Avg. Buy**	**Qty.**	**Last Close**	**Buy Value**
ASHOKLEY	₹148.90	0.03%	₹0.05	₹127.63	100	₹148.85	₹12,763.00
BHARTIARTL	₹789.50	-0.82%	-₹6.55	₹699.59	38	₹796.05	₹26,584.42
CAMS	₹2,089.90	0.06%	₹1.35	₹1,964.19	19	₹2,088.55	₹37,319.61
CDSL	₹994.80	1.08%	₹10.60	₹1,535.02	13	₹984.20	₹19,955.26
DIXON	₹2,912.00	0.68%	₹19.75	₹5,505.45	2	₹2,892.25	₹11,010.90
EXIDEIND	₹189.75	0.80%	₹1.50	₹205.80	47	₹188.25	₹9,672.60
FIEMIND	₹1,759.00	0.13%	₹2.30	₹1,206.35	21	₹1,756.70	₹25,333.35
IRFC	₹33.75	2.74%	₹0.90	₹26.00	575	₹32.85	₹14,950.00
KHAICHEM	₹75.40	-1.69%	-₹1.30	₹58.99	370	₹76.70	₹21,826.30
LATENTVIEW	₹337.50	0.69%	₹2.30	₹622.79	31	₹335.20	₹19,306.49
MAXHEALTH	₹482.00	0.39%	₹1.85	₹356.24	92	₹480.15	₹32,774.08
NMDC	₹108.10	-0.46%	-₹0.50	₹142.10	350	₹108.60	₹49,735.00
RADICO	₹1,110.95	-0.90%	-₹10.05	₹1,145.00	30	₹1,121.00	₹34,350.00
RUPA	₹257.10	1.30%	₹3.30	₹437.04	30	₹253.80	₹13,111.20
SAIL	₹84.05	0.12%	₹0.10	₹121.50	8	₹83.95	₹972.00
TATACHEM	₹988.50	1.44%	₹14.00	₹756.48	23	₹974.50	₹17,399.04
TATACOFFEE	₹230.00	0.15%	₹0.35	₹245.20	80	₹229.65	₹19,616.00
TATAPOWER	₹206.50	1.15%	₹2.35	₹102.64	600	₹204.15	₹61,584.00
ZOMATO	₹62.50	1.13%	₹0.70	₹158.50	94	₹61.80	₹14,899.00

Figure 9.6: informational-dashboard

9.3. Dashboard Data and Formats:

In the world of dashboards, the data used to populate them plays a vital role in ensuring their accuracy and relevance. Let's explore how dashboard data is sourced and the importance of choosing the right formats.

9.3.1. Sourcing Relevant and Up-to-Date Data

The foundation of a successful dashboard lies in sourcing relevant and up-to-date data. The data you choose should directly align with the purpose and objectives

of the dashboard. It should be error-free and authentic, as the information displayed on the dashboard will drive decisions, actions, and inferences. Depending on your needs, data can be obtained from various sources and formats such as spreadsheets, text files, web pages, or even organizational databases. By selecting the most appropriate data sources, you ensure that your dashboard presents a true reflection of your business's performance.

9.3.2. Medium and Format: Communicating Insights Effectively

While accurate data is crucial, the medium chosen for displaying the data is equally important. It's essential to select a format that effectively communicates insights and prevents any misleading impressions. The chosen data visualization should unambiguously present conclusions and drive the right understanding. Whether it's using charts, graphs, or other visual elements, the focus should be on clarity and precision in portraying the data.

Small multiples

Small multiples, also known as trellis charts or panel charts, are a series of charts that share the same scale and axes but display different subsets of data. By arranging multiple charts side by side, small multiples enable effective visual comparison of multiple categories or variables. They help identify patterns, trends, or variations across different segments, making it easier to understand the relationships within the data.

9.3.3. Live Data on Dashboards: Powering Real-Time Decision-Making

One of the remarkable advancements in dashboard technology is the ability to incorporate live data. With the advent of data warehousing and online analytical processing (OLAP), dashboards can now be refreshed instantly with real-time data. This feature has made dashboards increasingly sought after, not only by top management but also by regular users. It empowers decision-makers to access the most up-to-date information and make informed choices promptly. Additionally, the ability to refresh dashboards independently of the organization's IT department gives dashboard designers greater control and flexibility in obtaining and displaying data.

In the realm of Excel dashboards, understanding the significance of authentic and well-presented data is key to creating impactful visualizations. In the following sections, we'll explore practical techniques for sourcing, organizing, and presenting data effectively in Excel dashboards.

In the next subsection, we'll delve into the exciting world of visualizations and explore how to choose the most suitable chart types for your dashboard. So, let's continue our journey of creating visually stunning and insightful Excel dashboards that captivate and inform.

9.4. Excel Dashboards: Unleashing the Power of Excel's Features

When it comes to creating dashboards in Excel, the software offers a range of powerful features that bring data visualization to the forefront. In this section, we'll explore some of the most important Excel features that prove invaluable in designing impactful dashboards. These features enable you to simplify complex data and provide real-time visual insights into the current status or performance.

9.4.1. Excel Tables: The Backbone of Dashboard Data

The foundation of any dashboard lies in its data. Whether you're working with data from a single source or multiple sources, Excel tables provide an excellent way to organize and manage your data. With Excel's robust data import capabilities, you can establish connections to various data sources, allowing you to refresh your dashboard data whenever the source data updates. By naming your Excel tables, you can easily reference the data within your dashboard, simplifying the process and enhancing readability. These tables serve as your working tables, housing the raw data that forms the basis of your dashboard. Image 9.7

	A	B	C	D	E
1	% Profits Region-wise and Quarter-wise				
2		Qtr1	Qtr2	Qtr3	Qtr4
3	East	87%	90%	79%	96%
4	North	92%	94%	85%	97%
5	South	88%	95%	75%	80%
6	West	85%	87%	87%	88%

Figure 9.7: Table summary

9.4.2. Conditional Formatting: Highlighting Key Insights

Conditional formatting is a powerful tool that allows you to visually highlight specific data based on defined criteria. By applying formatting rules to your tables or cells, you can emphasize good and bad results, identify trends, and draw attention to critical information. This feature enables you to make your data more accessible and instantly discernible, guiding the viewer's focus towards key insights.

Holding Data	
Total Portfolio Value	₹443,162.25
Net Profit/Loss	-₹287,900.55
Current Portfolio Value	₹155,261.70
Yesterday's Value	₹516,916.35
Today's Profit/Loss	-₹361,654.65
Today's Profit/Loss %	-232.93%

Target Price Data	
Target Profit %	1639.39%
Target Profit	₹7,265,171.7
Net Target PF Value	**₹7,708,334.0**

Charges and Taxes (If Liquidated now)	
Depository Charge	₹302.67
Exchange txn. charge	₹20.65
GST on txn. charge	₹3.72
Stamp Duty	₹66.47
SEBI charge	₹0.60
Net Taxes & Charges (T&C)	**₹992.53**
T&C + P/L (Unrealised P/L)	**-₹288,893.08**
Unrealized P/L %	**-65.19%**
Unrealized PF Value	**₹154,269.17**
BEP %	**0.22%**
Profit reqd. for BEP	**₹992.53**

Figure 9.8: Conditional formatting

	A	B	C	D	E
1	*Holding Data*		*Charges and Taxes (If Liquidated now)*		
2	**Holdings**	19		Buy & Sell Turnc	₹5,98,423.95
3	**Total Portfolio Value**	₹4,43,162.25		Depository Char	₹302.67
4	**Net Profit/Loss %**	-64.97%		STT	₹598.42
5	**Net Profit/Loss**	⇩ (₹2,87,900.55)		Exchange txn. c	₹20.65
6	**Current Portfolio Value**	₹1,55,261.70		GST on txn. cha	₹3.72
7	**Yesterday's Value**	₹5,16,916.35		Stamp Duty	₹66.47
8	**Today's Profit/Loss**	⇩ (₹3,61,654.65)		SEBI charge	₹0.60
9	**Today's Profit/Loss %**	-232.93%		**Net Taxes & Ch**	**₹992.53**
10				**T&C + P/L (Unre**	⇩ (₹2,88,893.08)
11	*Target Price Data*		**Unrealized P/L**	**-65.19%**	
12	Target Profit %	1639.39%		**Unrealized PF V**	**₹1,54,269.17**
13	Target Profit	⇧ ₹72,65,171.75		**BEP %**	**0.22%**
14	**Net Target PF Value**	⇧ ₹77,08,334.00		**Profit reqd. for**	**₹992.53**

Figure 9.9: Conditional formatting

9.4.3. Charts and PivotTables: Summarizing and Visualizing Data

Charts and PivotTables are invaluable tools for summarizing and visualizing data in your dashboard. With a wide array of chart types to choose from, such

as bar charts, line charts, and pie charts, you can effectively showcase trends, comparisons, and distributions. PivotTables offer a dynamic way to summarize and analyze large datasets, allowing you to extract key information and present it in a meaningful and interactive format. These visual elements enable you to transform complex data into easily understandable visual representations, making your dashboard visually appealing and informative.

Row Labels	Sum of Profit	Sum of Revenue	Sum of Cost
Jan	₹ 1,71,318	₹ 2,91,853	₹ 1,20,535
Feb	₹ 1,35,968	₹ 2,31,925	₹ 95,957
Mar	₹ 1,29,483	₹ 2,19,071	₹ 89,588
Apr	₹ 1,96,766	₹ 3,33,380	₹ 1,36,614
May	₹ 1,26,857	₹ 2,15,866	₹ 89,009
Jun	₹ 2,50,815	₹ 4,26,848	₹ 1,76,033
Jul	₹ 1,73,934	₹ 2,96,171	₹ 1,22,237
Aug	₹ 1,53,357	₹ 2,61,755	₹ 1,08,398
Sep	₹ 2,65,778	₹ 4,50,850	₹ 1,85,072
Oct	₹ 4,84,487	₹ 8,21,032	₹ 3,36,545
Nov	₹ 2,89,837	₹ 4,89,991	₹ 2,00,154
Dec	₹ 3,84,765	₹ 6,51,577	₹ 2,66,812
Grand Total	**₹ 27,63,364**	**₹ 46,90,319**	**₹ 19,26,955**

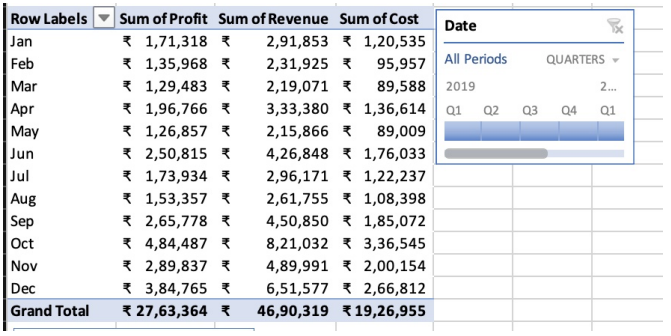


Figure 9.10: pivot tables

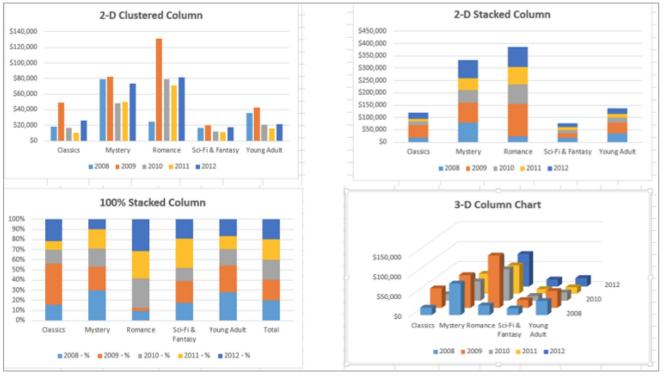


Figure 9.11: charts

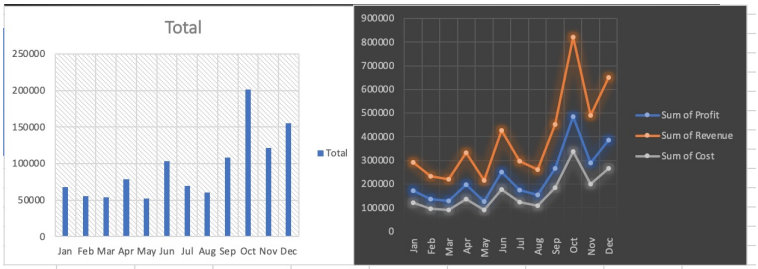


Figure 9.12: charts

9.4.4. Interactive Controls: Empowering User Interaction

Interactivity is a key aspect of modern dashboards, and Excel provides various interactive controls to enhance user experience. You can add elements like slicers, drop-down menus, and buttons, enabling users to filter, drill down, and explore the data according to their needs. These interactive controls empower users to interact with the dashboard dynamically, gaining deeper insights and customizing their viewing experience. Image 9.10

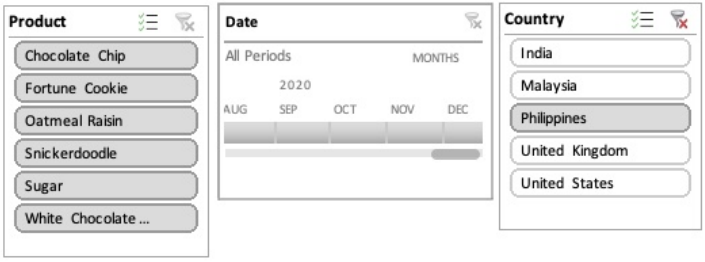


Figure 9.13: Interactive elements.

9.4.5. KPI Management: Monitoring Performance Indicators

Key Performance Indicators (KPIs) play a crucial role in monitoring and assessing business performance. Excel offers tools for defining, managing, and displaying KPIs within your dashboard. By setting thresholds and creating visual indicators, you can easily track progress, identify areas of concern, and align your dashboard with organizational objectives. The inclusion of KPIs enhances the effectiveness of your dashboard as a performance monitoring tool.

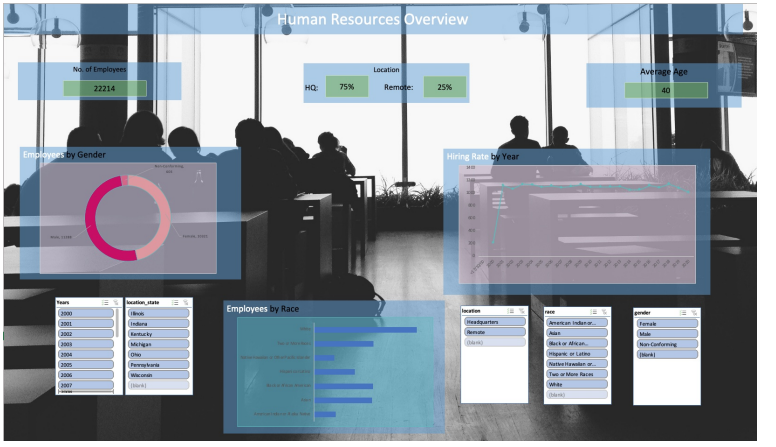


Figure 9.14: kpi dashboard

9.4.6. Sparklines: Unleashing Trends in Compact Charts

When it comes to showcasing trends over time within your Excel tables, Sparklines come to the rescue. Sparklines are mini charts that can be placed in single cells, allowing you to visualize trends in a compact and concise format. With Sparklines, you can choose from line charts, column charts, or win-loss charts to depict the trends based on your data.

Sparklines provide a quick and efficient way to convey trends and patterns within a small space, making them an ideal choice for embedding within your dashboard tables. By adding Sparklines to your Excel tables, you can instantly visualize the ups and downs, progress, or fluctuations in your data. These compact charts offer a condensed view of trends and provide valuable insights at a glance.

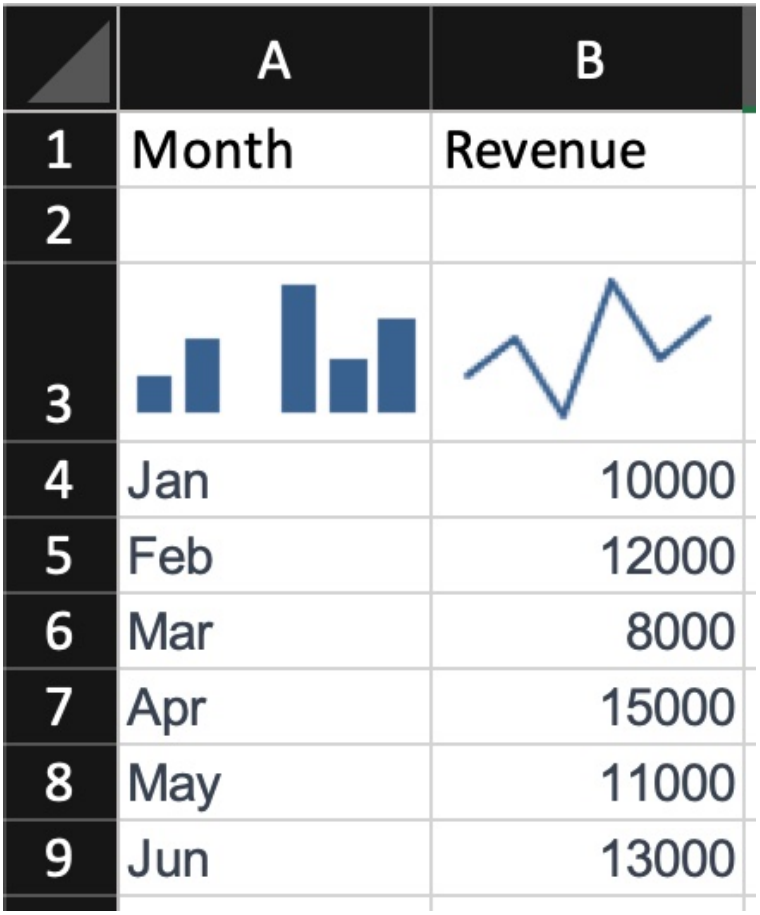


Figure 9.15: Spark lines

10

Cleaning and Manipulating Data

In this Chapter

In this chapte, you will discover how macros can be employed to automate an extensive range of tasks in Excel, leading to improved efficiency and ensuring consistent execution of repetitive tasks. By utilizing Excel's macro recorder to capture routine tasks, users can complete tasks more quickly and maintain a consistent process each time the task is performed.

It is important to understand that the macro recorder uses the Visual Basic for Applications (VBA) language, a specialized version of the BASIC programming language developed by Microsoft for Office applications.

Learners will also explore using Excel's Visual Basic Editor to view and modify the VBA code of recorded macros. Throughout this chapter, learners will gain practical knowledge on how to record, test, and play back macros for automating repetitive tasks involved in creating and using Excel worksheets and charts.

Macros and automation in Excel

Macros are sequences of instructions that automate repetitive tasks or processes, allowing you to save time and improve efficiency. By recording or writing macros, you can automate various operations, such as data entry, formatting, calculations, and more. Understanding macros and automation empowers you to streamline workflows and perform complex tasks with just a few clicks.

10.1. What are macros?

Macros are a set of automated actions that allow users to execute repetitive tasks in Excel more efficiently. They are written using Visual Basic for Applications (VBA), a programming language designed specifically for Microsoft Office applications.

There are two methods to create macros in Excel:

- Utilize Excel's macro recorder to capture your actions as you perform them in a worksheet.
- Input the desired instructions directly as VBA code in the Visual Basic Editor.

10.1.1. Macros : Optimizing the Datascience

Macros can be used in data science to:

Data cleaning and preprocessing: Macros can be used to automate tasks such as removing duplicates, handling missing values, standardizing data formats, and converting data types.

Data transformation: Macros can assist in aggregating data, splitting or merging columns, reshaping data, and performing calculations to create new variables.

Importing and exporting data: Automate the process of importing data from external sources like text files, CSV files, databases, and other workbooks, as well as exporting data to different formats such as CSV, PDF, or other file types.

Generating summary statistics: Create macros to calculate and update summary statistics like mean, median, mode, standard deviation, and percentiles, ensuring consistency across analyses.

Automating report generation: Macros can be used to generate dynamic reports and dashboards, automatically updating data and visualizations based on user-defined criteria.

Creating custom functions: Develop custom VBA functions tailored to specific data science needs, enabling more advanced data manipulation and analysis.

Automating data visualization: Macros can help automate the creation and customization of charts and graphs, such as bar charts, line charts, pie charts, and scatter plots, ensuring consistent visual representation of data.

Macros

Macros are sets of instructions or code written in Visual Basic for Applications (VBA) that automate repetitive tasks in Excel. They allow you to record a series of actions or write custom code to perform specific operations. Macros can be triggered by buttons, shortcuts, or events, providing an efficient way to automate tasks and enhance productivity in Excel.

10.1.2. Enabling the Developer tab in Excel

Excel allows you to include a Developer tab in the Ribbon, which features a Record Macro button and a Use Relative References button, both essential for recording macros.

To enable the Developer tab in Excel, follow these steps:

1. Click on the "File" tab, and then click "Options." <image 10.1>
2. In the Excel Options dialog box, select "Customize Ribbon." <image 10.2>
3. Under the "Customize the Ribbon" or “View” list, check the "Developer" option. <image 10.3>
4. Click "OK" to apply the changes.

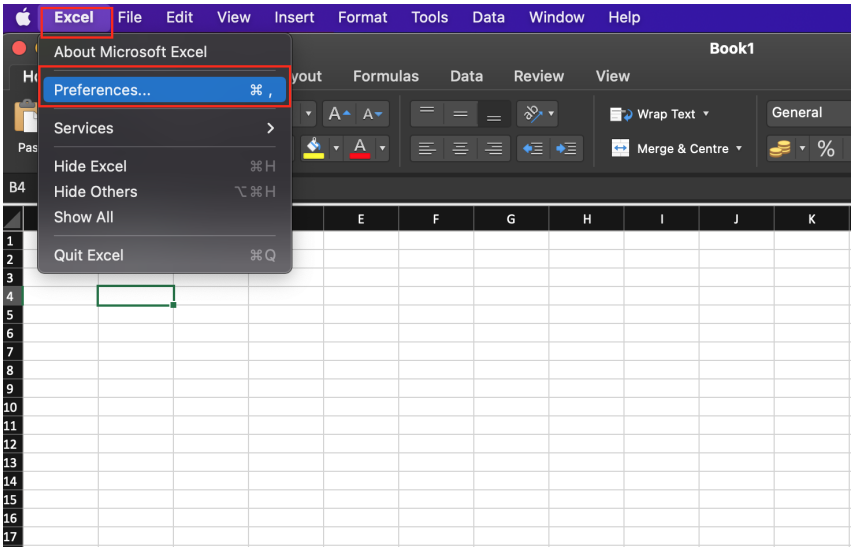


Figure 10.1: Enabling developer mode

10.2. Recording a Macro

Macros table

10.2.1. The Record Macro dialog box

To access the Record Macro dialog box, follow these steps:

1. Navigate to the "Developer" tab.
2. Click on "Record Macro." Macro name
3. Assign a unique, descriptive name to the macro.
4. Avoid using spaces and special characters.
5. Begin the macro name with a letter. Shortcut key <image 10.5>
6. Assign a shortcut key to run the macro quickly.

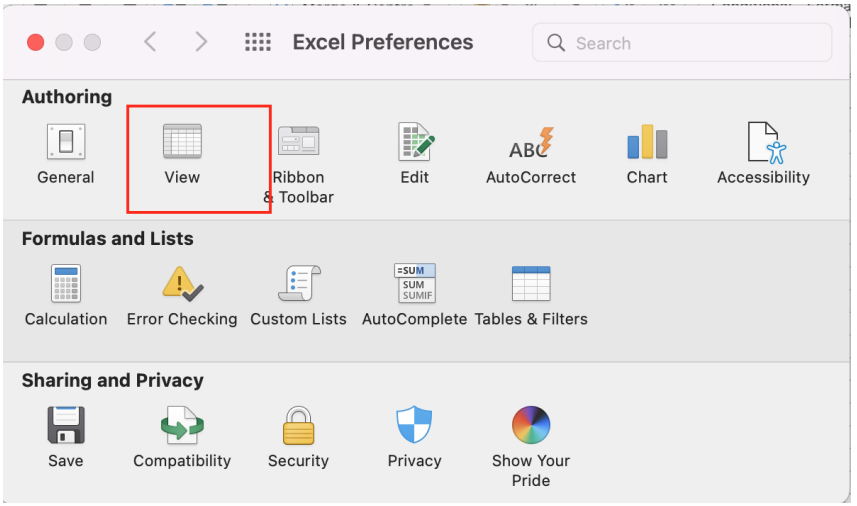


Figure 10.2: Click on view to access the option

10.2.2. Recording a simple macro

- Open the Record Macro dialog box.
- Enter a macro name, shortcut key, storage location, and description.
- Perform the actions you want the macro to automate.
- Click "Stop Recording" on the Developer tab.

10.2.3. Running the recorded macro

Use the assigned shortcut key or open the Macro dialog box (Developer tab > Macros) to run the macro.

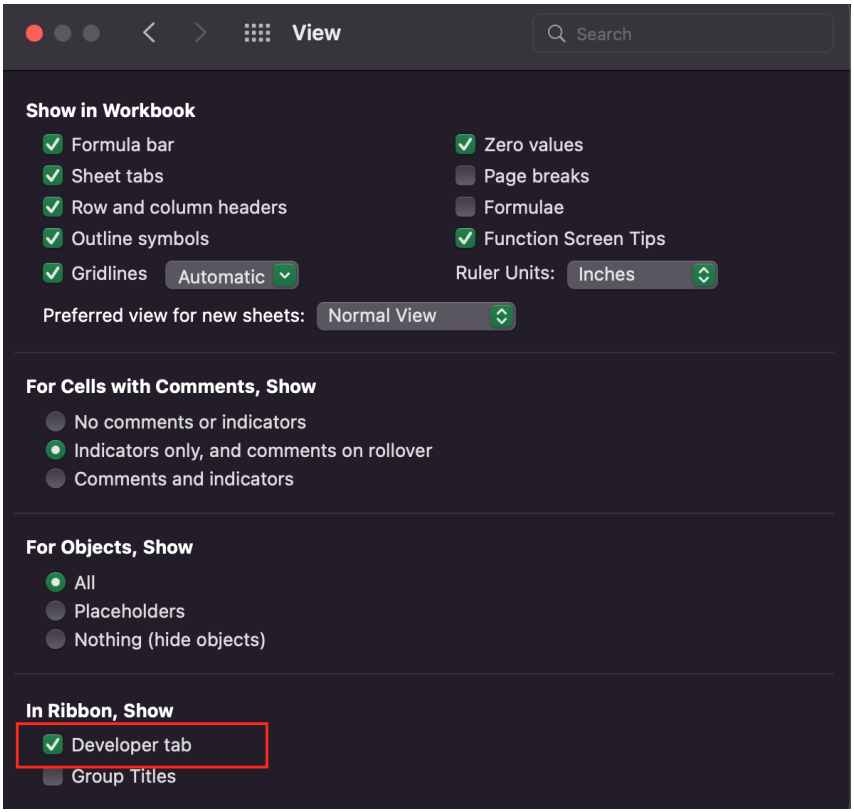
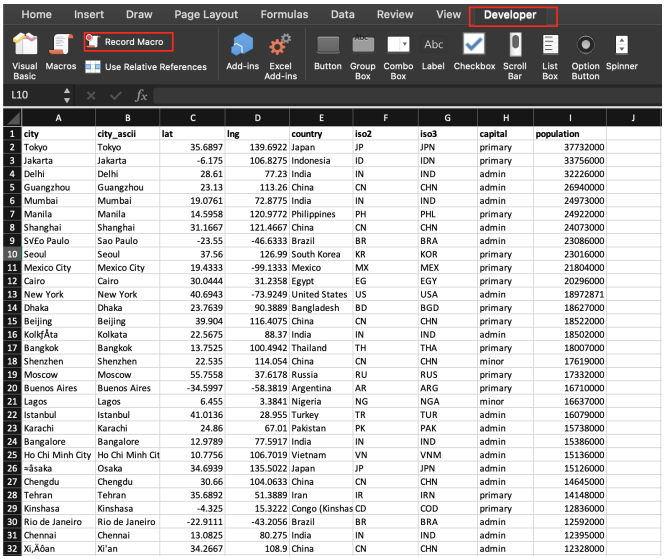


Figure 10.3: check mark developer tab in order to enable it

10.3. Effortless Automation : Visual Basic for Applications (VBA)

Visual Basic for Applications (VBA) is an event-driven programming language integrated into Microsoft Office applications, like Excel. VBA enables you to automate tasks, manipulate data, and customize your user experience within Excel. In this guide, we will cover the fundamentals of VBA, including its components, syntax, and essential techniques to help you harness its full potential.



	A	B	C	D	E	F	G	H	I	J
1	city	city_ascii	lat	lng	country	iso2	iso3	capital	population	
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	primary	37732000	
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	primary	33756000	
4	Delhi	Delhi	28.61	77.23	India	IN	IND	admin	32226000	
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	admin	26940000	
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	admin	24973000	
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	primary	24922000	
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	admin	24073000	
9	SVEo Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	admin	23086000	
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	primary	23016000	
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	primary	21804000	
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	primary	20296000	
13	New York	New York	40.6943	-73.9249	United States	US	USA	admin	18972871	
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	primary	18627000	
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	primary	18522000	
16	KolkÍÅta	Kolkata	22.5675	88.37	India	IN	IND	admin	18502000	
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	primary	18007000	
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	minor	17619000	
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	primary	17332000	
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	primary	16710000	
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	minor	16637000	
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	admin	16079000	
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	admin	15738000	
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	admin	15386000	
25	Ho Chi Minh City	Ho Chi Minh Cit	10.7756	106.7019	Vietnam	VN	VNM	admin	15136000	
26	=åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	admin	15126000	
27	Chengdu	Chengdu	30.66	104.0633	China	CN	CHN	admin	14645000	
28	Tehran	Tehran	35.6892	51.3889	Iran	IR	IRN	primary	14148000	
29	Kinshasa	Kinshasa	-4.325	15.3222	Congo (Kinshas	CD	COD	primary	12836000	
30	Rio de Janeiro	Rio de Janeiro	-22.9111	-43.2056	Brazil	BR	BRA	admin	12592000	
31	Chennai	Chennai	13.0825	80.275	India	IN	IND	admin	12395000	
32	Xi,Âôan	Xi'an	34.2667	108.9	China	CN	CHN	admin	12328000	

Figure 10.4: Select Developer tab to access the Macros fucntionality

10.3.1. VBA Environment and Components

Visual Basic Editor (VBE)

VBE is the integrated development environment (IDE) for VBA, which can be accessed by pressing Alt + F11 in Excel. It provides a workspace for writing, editing, and debugging VBA code.

Projects, Modules, and Procedures

In VBE, a VBA project corresponds to a workbook, and it contains various components like modules, user forms, and class modules. A module is a container for VBA code, while procedures (Subs and Functions) are blocks of code that perform specific tasks.

VBA syntax and structure

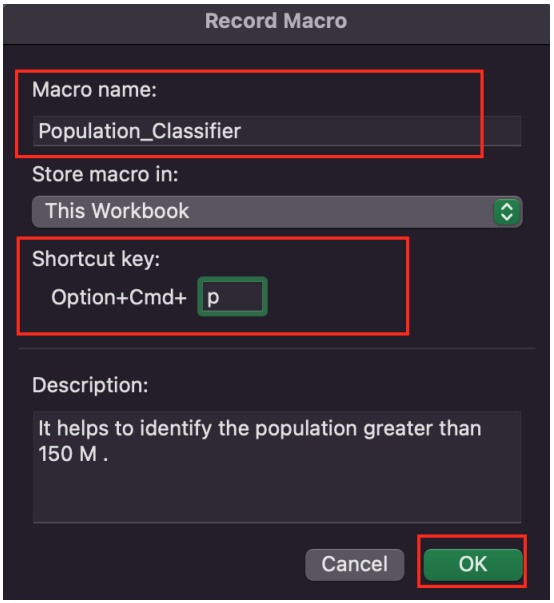


Figure 10.5: Recording it assign a shortcut key to trigger macros

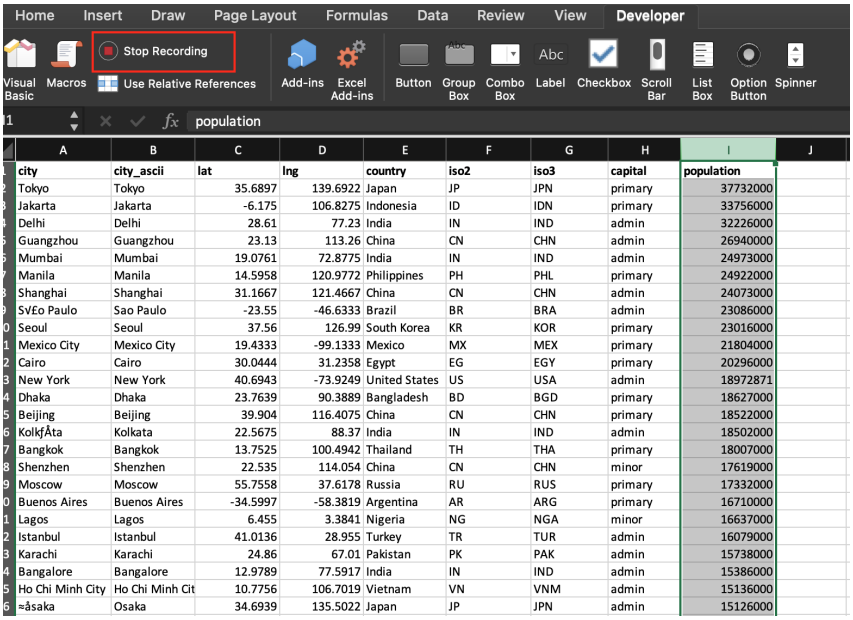
Variables and Data Types

Variables store data in memory and must be declared using the Dim keyword, followed by the variable name and data type (e.g., Dim varName As DataType).

Common data types include

1. Numeric data types (Byte, Integer, Long, Single, Double, and Currency)
2. Text data types (String)
3. Date and time data types (Date)
4. Other data types (Boolean, Object, and Variant)

Control Structures



	A	B	C	D	E	F	G	H	I	J
1	city	city_ascii	lat	lng	country	iso2	iso3	capital	population	
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	primary	37732000	
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	primary	33756000	
4	Delhi	Delhi	28.61	77.23	India	IN	IND	admin	32226000	
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	admin	26940000	
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	admin	24973000	
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	primary	24922000	
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	admin	24073000	
9	SV£o Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	admin	23086000	
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	primary	23016000	
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	primary	21804000	
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	primary	20296000	
13	New York	New York	40.6943	-73.9249	United States	US	USA	admin	18972871	
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	primary	18627000	
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	primary	18522000	
16	Kolkƒåta	Kolkata	22.5675	88.37	India	IN	IND	admin	18502000	
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	primary	18007000	
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	minor	17619000	
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	primary	17332000	
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	primary	16710000	
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	minor	16637000	
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	admin	16079000	
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	admin	15738000	
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	admin	15386000	
25	Ho Chi Minh City	Ho Chi Minh Cit	10.7756	106.7019	Vietnam	VN	VNM	admin	15136000	
26	≈åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	admin	15126000	

Figure 10.6: Perform step to sort the population in desceding order and record it

Control structures are essential for managing the flow of your VBA programs. They allow you to make decisions, perform repetitive tasks, and control the execution of your code. In this guide, we will discuss various control structures available in VBA, including:

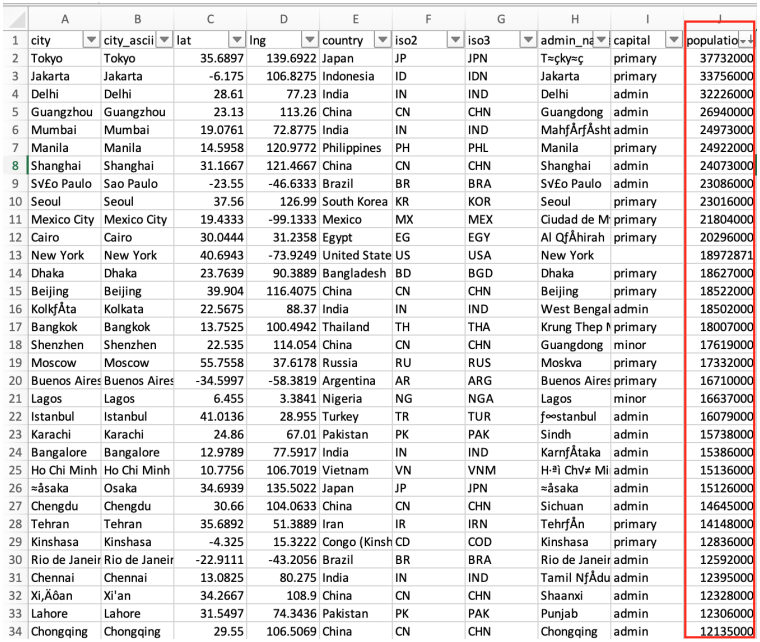
Conditional Statements

Loop Statements

Exit and GoTo Statements

Conditional Statements

Conditional statements in VBA enable you to execute specific blocks of code based on certain conditions.



	A	B	C	D	E	F	G	H	I	J
1	city	city_ascii	lat	lng	country	iso2	iso3	admin_na	capital	populatio
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	T=çky=ç	primary	37732000
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	Jakarta	primary	33756000
4	Delhi	Delhi	28.61	77.23	India	IN	IND	Delhi	admin	32226000
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	Guangdong	admin	26940000
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	MahfÂrfÂsht	admin	24973000
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	Manila	primary	24922000
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	Shanghai	admin	24073000
9	SV£o Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	SV£o Paulo	admin	23086000
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	Seoul	primary	23016000
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	Ciudad de M	primary	21804000
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	Al QfÂhirah	primary	20296000
13	New York	New York	40.6943	-73.9249	United State	US	USA	New York		18972871
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	Dhaka	primary	18627000
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	Beijing	primary	18522000
16	KolkƒÂta	Kolkata	22.5675	88.37	India	IN	IND	West Bengal	admin	18502000
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	Krung Thep M	primary	18007000
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	Guangdong	minor	17619000
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	Moskva	primary	17332000
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	Buenos Aires	primary	16710000
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	Lagos	minor	16637000
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	ƒ∞stanbul	admin	16079000
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	Sindh	admin	15738000
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	KarnƒÂtaka	admin	15386000
25	Ho Chi Minh	Ho Chi Minh	10.7756	106.7019	Vietnam	VN	VNM	H·ªì Chí Mi	admin	15136000
26	≈åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	≈åsaka	admin	15126000
27	Chengdu	Chengdu	30.66	104.0633	China	CN	CHN	Sichuan	admin	14645000
28	Tehran	Tehran	35.6892	51.3889	Iran	IR	IRN	TehrƒÂn	primary	14148000
29	Kinshasa	Kinshasa	-4.325	15.3222	Congo (Kinsh	CD	COD	Kinshasa	primary	12836000
30	Rio de Janeir	Rio de Janeir	-22.9111	-43.2056	Brazil	BR	BRA	Rio de Janeir	admin	12592000
31	Chennai	Chennai	13.0825	80.275	India	IN	IND	Tamil NƒÂdu	admin	12395000
32	Xi,ÂÒan	Xi'an	34.2667	108.9	China	CN	CHN	Shaanxi	admin	12328000
33	Lahore	Lahore	31.5497	74.3436	Pakistan	PK	PAK	Punjab	admin	12306000
34	Chongqing	Chongqing	29.55	106.5069	China	CN	CHN	Chongqing	admin	12135000

Figure 10.7: Automation for sorting the values

We will cover two main types of conditional statements in VBA:

A. If...Then...Else Statements

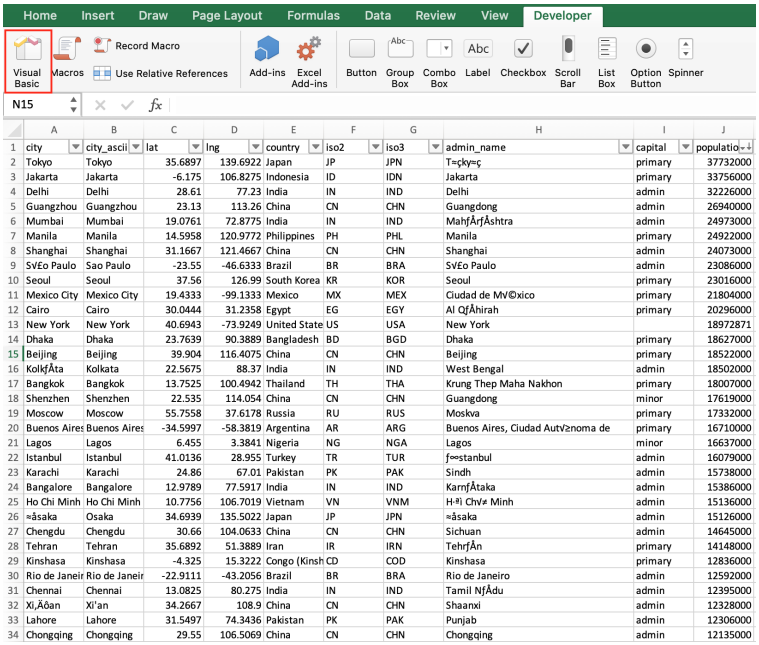
The If...Then...Else statement allows you to execute different blocks of code based on one or multiple conditions. It comes in several forms:

Simple If...Then statement

If...Then...Else statement

If...Then...ElseIf statement

Nested If...Then...Else statement



	A	B	C	D	E	F	G	H	I	J
1	city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	populatio
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	T=çky=ç	primary	37732000
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	Jakarta	primary	33756000
4	Delhi	Delhi	28.61	77.23	India	IN	IND	Delhi	admin	32226000
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	Guangdong	admin	26940000
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	MahƒÁrƒÁshtra	admin	24973000
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	Manila	primary	24922000
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	Shanghai	admin	24073000
9	SvÉo Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	SvÉo Paulo	admin	23086000
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	Seoul	primary	23016000
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	Ciudad de MV©xico	primary	21804000
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	Al QƒÂhirah	primary	20296000
13	New York	New York	40.6943	-73.9249	United State	US	USA	New York		18972871
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	Dhaka	primary	18627000
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	Beijing	primary	18522000
16	KolkƒÂta	Kolkata	22.5675	88.37	India	IN	IND	West Bengal	admin	18502000
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	Krung Thep Maha Nakhon	primary	18007000
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	Guangdong	minor	17619000
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	Moskva	primary	17332000
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	Buenos Aires, Ciudad AutV2noma de	primary	16710000
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	Lagos	minor	16637000
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	f∞stanbul	admin	16079000
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	Sindh	admin	15738000
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	KarnƒÂtaka	admin	15386000
25	Ho Chi Minh	Ho Chi Minh	10.7756	106.7019	Vietnam	VN	VNM	H·ª‰ Chv≠ Minh	admin	15136000
26	≈åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	≈åsaka	admin	15126000
27	Chengdu	Chengdu	30.66	104.0633	China	CN	CHN	Sichuan	admin	14645000
28	Tehran	Tehran	35.6892	51.3889	Iran	IR	IRN	TehrƒÂn	primary	14148000
29	Kinshasa	Kinshasa	-4.325	15.3222	Congo (Kinsh	CD	COD	Kinshasa	primary	12836000
30	Rio de Janeir	Rio de Janeir	-22.9111	-43.2056	Brazil	BR	BRA	Rio de Janeiro	admin	12592000
31	Chennai	Chennai	13.0825	80.275	India	IN	IND	Tamil NƒÂdu	admin	12395000
32	Xi¸Âan	Xi'an	34.2667	108.9	China	CN	CHN	Shaanxi	admin	12328000
33	Lahore	Lahore	31.5497	74.3436	Pakistan	PK	PAK	Punjab	admin	12306000
34	Chongqing	Chongqing	29.55	106.5069	China	CN	CHN	Chongqing	admin	12135000

Figure 10.8: Accessing Visual Basic Editor

B. Select Case Statements

The Select Case statement is an alternative to multiple

If...Then...ElseIf statements. It is used when you need to test a single expression against multiple values. This control structure makes your code easier to read and maintain.

Loop Statements

Loop statements are used to execute a block of code repeatedly based on specific conditions.

VBA offers four types of loop structures:

1. **For...Next Loops**

   For...Next loops are used when you know how many times you want to repeat a block of code. They use a counter variable that increments or decrements with each iteration.

2. **For Each...Next Loops**

   For Each...Next loops are used to iterate through the elements of a collection or an array. They are useful for processing each item in a collection without knowing its exact size.

3. **Do...Loop Statements**

   Do...Loop statements are used when you want to repeat a block of code while a certain condition is true or until a specific condition becomes true. There are four variations of Do...Loop statements:

   Do While...Loop

   Do Until...Loop

   Loop While...

   Loop Until...

4. **While...Wend Loops**

   While...Wend loops are used to repeat a block of code as long as a given condition is true. This loop structure is less flexible than the Do...Loop statement and is not recommended for new projects.

Exit and GoTo Statements

A. Exit Statement

The Exit statement is used to prematurely exit a loop or a procedure when a specific condition is met. It can be used with For...Next loops, For Each...Next

loops, Do...Loop statements, and procedures (functions and subroutines).

B. GoTo Statement

The GoTo statement allows you to jump to a specific line in your code. Although it can be useful in some error handling scenarios, it is generally discouraged due to its potential to create complex and hard-to-maintain code.

Procedures

Procedures are blocks of code that perform specific tasks.

Function:

Functions are procedures that return a value. They are used when you need to perform a specific task and obtain a result, which can be used in other parts of your code or within Excel formulas. Functions can be built-in, like VBA's MsgBox or Excel's VLOOKUP, or user-defined.

Creating function:

To create a function in VBA, use the following syntax:

Function FunctionName(Arguments) As ReturnType

' Code to be executed

FunctionName = ReturnValue

End Function

Subroutines:

Subroutines, also known as Sub procedures, do not return a value. They are used to perform a specific task or a series of tasks, such as automating repetitive actions, formatting data, or generating reports. Subroutines are typically triggered by events or called from other procedures.

Subroutines

Subroutines are blocks of code that perform a specific task or set of tasks. They are used to organize and modularize code, making it easier to read, understand, and maintain. Subroutines (Sub) do not return the value in the respective code of block, it only performs the required tasks.

Creating subroutines :

To create a subroutine in VBA, use the following syntax:

Sub SubroutineName(Arguments) ' Code to be executed End Sub

Subroutines (Sub) do not return a value, while functions (Function) return a value and can be used in formulas.

10.3.2. Editing recorded Macros in VBA

Once you've created a macro, you don't always need to re-record it to modify its behavior. In many situations, it is more efficient to alter the macro's actions by editing its content within the Visual Basic Editor. Keep in mind that if the macro you wish to edit is located in your Personal Macro Workbook . you must first unhide this workbook before proceeding with edits in the Visual Basic Editor.

To edit a macro in the Visual Basic Editor, follow these steps:

1. Select either Developer → Macros (or press Alt+LPM) or View →
2. Macros → View Macros (or press Alt+WMV). You can also use the Alt+F8 shortcut.
3. Excel displays the Macro dialog box, which lists the names of all macros

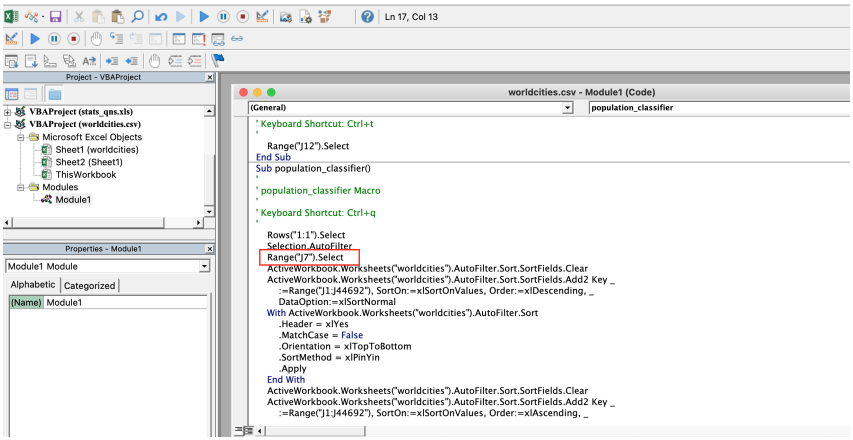


Figure 10.9: Macros Editing can be done using VBA interface

defined in the active workbook, any other open workbooks, and your Personal Macro Workbook.

4. Locate the macro you wish to edit in the Macro Name list box, and then click the Edit button.

5. Excel launches the Visual Basic Editor with your macro's code visible in the Code window. This will not occur if you have chosen a macro stored in the Personal Macro Workbook and that workbook remains hidden.

In that case, Excel displays an Alert dialog box telling you that you can't edit a hidden macro and informing you that you need to unhide this workbook. You then need to click OK in the Alert dialog box, press Escape to close the Macro dialog box, and then follow the steps for unhiding the Personal Macro Workbook immediately preceding these steps before you repeat these first two macro editing steps. After you have the lines of code for the macro displayed in the Code window in the Visual Basic Editor, you can edit any of its statements as needed. When editing the macro's commands, remember that you can undo any deletion that you make by mistake (Edit → Undo; Ctrl+Z).

Edit the statements in the Code window of the Visual Basic Editor as needed. After you finish editing the macro, you're ready to return to your worksheet, where you can test the modified macro and make sure that you haven't added some wacky, unwanted command to the macro or, even worse, crippled it so that it no longer runs at all.

Click the View Microsoft Excel button at the beginning of the Standard toolbar (see Figure 2-2) or click the workbook's minimized button on the Windows taskbar. Select an appropriate or safe place in which to test your modified macro and then run it, either by pressing its shortcut keys or by pressing Alt+F8, clicking it in the Macro list box, and then clicking the Run button. If something doesn't work as intended or if the macro doesn't work at all, you need to return to the Visual Basic Editor and find and correct your error(s). Choose Developer → Visual Basic (or press Alt+LV) to return to the Visual Basic Editor and have a try at editing the code one more time. If everything checks out and runs as planned, you need to save your changes as outlined in Step 5.

Click the Save button on the Standard toolbar to save the changes to the modified macro if it's stored as part of the current workbook

10.4. Finding and replacing code in the macro

When working with Excel macros, you may need to find and replace specific code elements to update or improve the functionality of your macro. Excel's Visual Basic for Applications (VBA) editor offers a Find and Replace feature that makes this process easy and efficient.

Follow these steps to find and replace code in Excel macros:

1. **Open the Visual Basic Editor:** First, open the Excel workbook containing the macro you want to modify. Press Alt + F11 to open the Visual Basic Editor, or navigate to the Developer tab and click on the "Visual Basic" button.

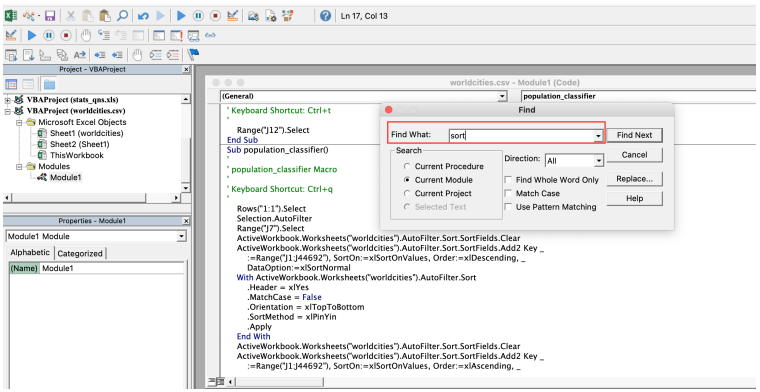


Figure 10.10: Accessing Find and Replace functionality in VBA by pressing Ctrl+F

2. **Locate the macro:** In the Visual Basic Editor's Project Explorer window (usually located on the left side), expand the folder containing your macro (e.g., VBAProject (YourWorkbookName.xlsm)). Then, double-click the module containing the macro to display its code in the Code window.

3. **Open the Find and Replace dialog box:** In the Code window, press Ctrl + F to open the Find and Replace dialog box. You can also navigate to the "Edit" menu and select "Find" or "Replace" to access the same dialog box.

4. **Find specific code:** In the Find and Replace dialog box, enter the code snippet you want to find in the "Find What" field. You can choose additional options such as "Match Case" or "Match Whole Word" to refine your search. Click "Find Next" to locate the first occurrence of the specified code. Continue clicking "Find Next" to navigate through all instances of the code.

5. **Replace code:** To replace the found code with a new code snippet, switch to the "Replace" tab within the Find and Replace dialog box. Enter the new code in the "Replace With" field, and click "Replace" to replace the currently selected code instance or "Replace All" to replace all occurrences of the specified code.

6. **Review and save changes:** Carefully review the changes made to your macro code in the Code window. Ensure that the new code is correct and does not introduce any errors. If you are satisfied with the modifications, save the changes by clicking the "Save" button on the toolbar or by pressing Ctrl + S.

7. **Test the updated macro:** Return to Excel by pressing Alt + Q or clicking the "View Microsoft Excel" button on the toolbar. Test the modified macro to ensure it functions as intended. If any issues arise, revisit the Visual Basic Editor and make the necessary adjustments to the code.

10.5. Writing new Macros in the visual Basic Editor

After mastering VBA basics, you can create new macros in the Visual Basic Editor instead of only modifying pre-recorded ones in Excel.

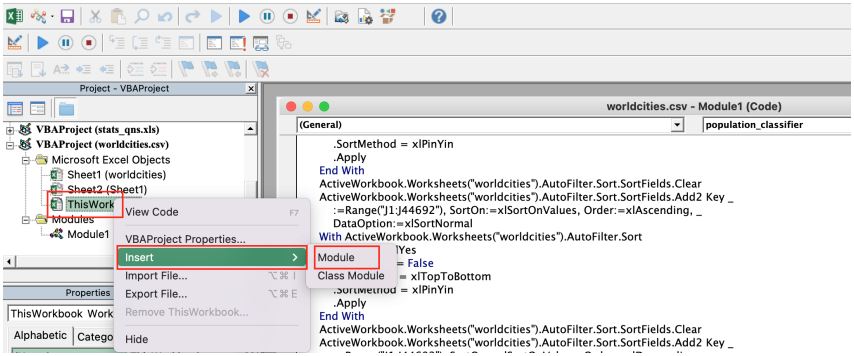


Figure 10.11: Adding a Module for writing a new Macros

To create a macro from scratch, follow these steps:

1. In the Project Explorer, click the VBA project where you want the new

macro. For the current workbook, click VBAProject with the filename in parentheses, e.g., VBAProject (My Worksheet). For a global macro, click VBAProject (PERSONAL.XLSB).

2. Select Insert → Module from the Visual Basic Editor menu bar. Excel opens a blank Code window and adds a Module icon with the next available number in the Project Explorer.Type "sub" (short for subprocedure) and press the spacebar.

3. Enter your macro's name and press Enter, adhering to range name rules (start with a letter, no spaces). The Visual Basic Editor adds parentheses, a blank line, and an End Sub statement, positioning the insertion point between the Sub and End Sub lines.

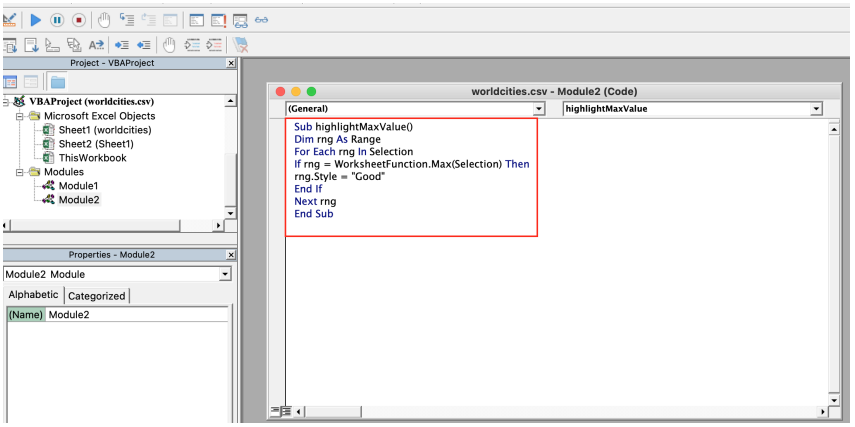


Figure 10.12: Writing Macros for highlighting the Max value in the data

4. Write your VBA code between the Sub and End Sub statements. Document the macro's purpose with comments, starting each line with an apostrophe. For actual VBA statements, don't use apostrophes. Use Tab to indent and Shift+Tab to outdent lines for readability. Consult VBA online help for assistance.

5. Save your macro using File → Save or Ctrl+S in the Visual Basic Editor.

Return to your worksheet and run the new macro with Developer → Macros or View → Macros (or Alt+F8), selecting the macro name and clicking OK.

10.6. Running Macros in the visual Basic Editor

While running macros from Excel's Macro dialog box is usually preferred, you can also execute them within the Visual Basic Editor.

Follow these steps:

- **Step 1:** If necessary, switch to the worksheet for your macro's changes and select the starting cell.
- **Step 2:** Return to the Visual Basic Editor.
- **Step 3:** Click within the macro you want to execute.
- **Step 4: To run the macro, use either of these methods:**

    1. For running the entire macro, select Run → Run Sub/UserForm, click the Standard toolbar's Run Sub/UserForm button (as shown in Figure), or press F5.
    2. To step through the macro line by line (helpful for testing each line's impact on your worksheet), choose Debug → Step Into or press F8. Use this command (F8 is recommended) to execute one line at a time. To finish the macro without stopping, press F5; to halt the macro, select Run → Reset or click the Standard toolbar's Reset button.

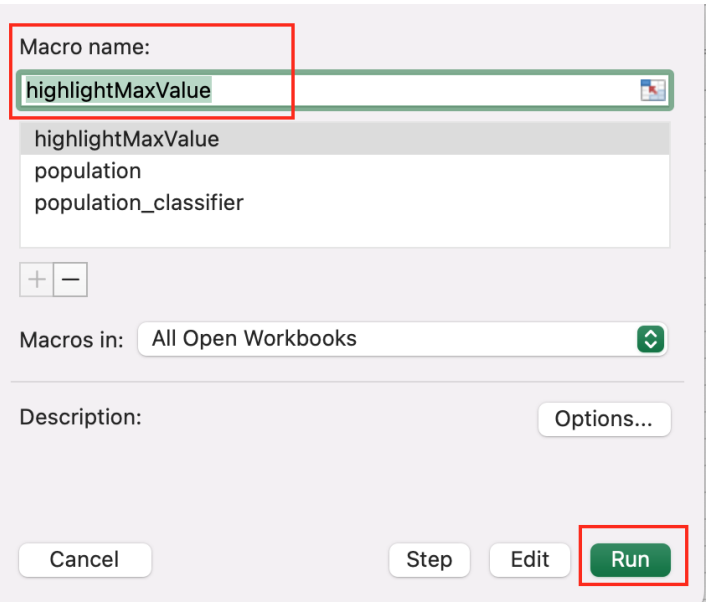


Figure 10.13: New Created Macros named as highlight max value

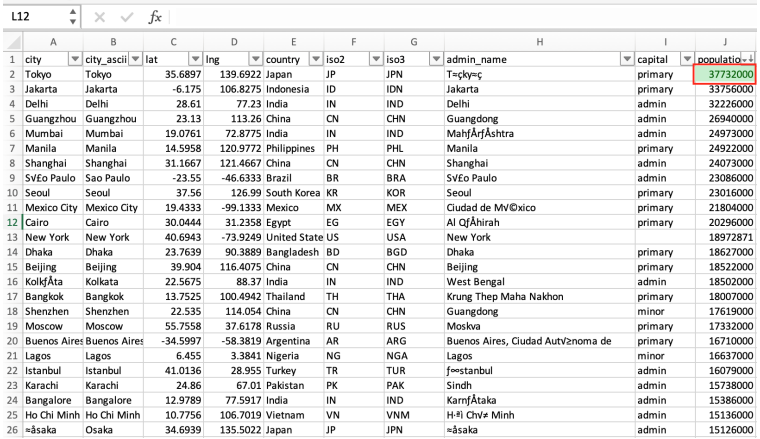
10.7. Creating Custom Formulas in Excel

Calculate table

VBA in Excel is excellent for creating custom worksheet functions, also known as user-defined functions (UDFs). UDFs are convenient because they don't require the Macro dialog box for execution. Instead, you can input them into your worksheets like any built-in Excel function, using the Insert Function button on the Formula bar or typing them directly into a cell.

To create a UDF, follow these four steps:

1. In the Visual Basic Editor, create a new module sheet for the custom function by selecting its project in the Project Explorer window and choosing Insert → Module from the menu bar.



	A	B	C	D	E	F	G	H	I	J
1	city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	populatio
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	T=çky=ç	primary	37732000
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	Jakarta	primary	33756000
4	Delhi	Delhi	28.61	77.23	India	IN	IND	Delhi	admin	32226000
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	Guangdong	admin	26940000
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	MahƒÁrƒÅshtra	admin	24973000
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	Manila	primary	24922000
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	Shanghai	admin	24073000
9	SVÉo Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	SVÉo Paulo	admin	23086000
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	Seoul	primary	23016000
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	Ciudad de MV©xico	primary	21804000
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	Al QƒÂhirah	primary	20296000
13	New York	New York	40.6943	-73.9249	United State	US	USA	New York		18972871
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	Dhaka	primary	18627000
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	Beijing	primary	18522000
16	KolkƒÂta	Kolkata	22.5675	88.37	India	IN	IND	West Bengal	admin	18502000
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	Krung Thep Maha Nakhon	primary	18007000
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	Guangdong	minor	17619000
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	Moskva	primary	17332000
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	Buenos Aires, Ciudad AutV≥noma de	primary	16710000
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	Lagos	minor	16637000
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	ƒ∞stanbul	admin	16079000
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	Sindh	admin	15738000
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	KarnƒÂtaka	admin	15386000
25	Ho Chi Minh	Ho Chi Minh	10.7756	106.7019	Vietnam	VN	VNM	H·ªì Chí Minh	admin	15136000
26	≈åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	≈åsaka	admin	15126000

Figure 10.14: Highlighted highest value for the population

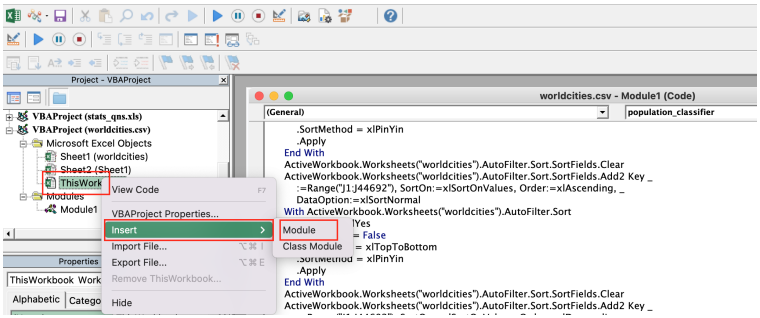


Figure 10.15: Adding Module to define a New function

2. In the Code window, enter the custom function's name and specify its argument names in parentheses on the first line. Avoid duplicating built-in function names (e.g., SUM, AVERAGE) and list argument names in the order they are processed.
3. Write the formula or formulas that instruct Excel on calculating the custom function's result using the argument names listed in the Function command. Include necessary arithmetic operators or built-in functions on the line(s) below.

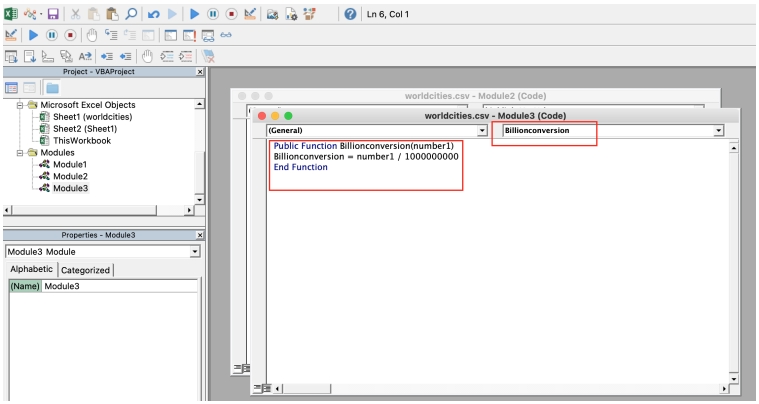
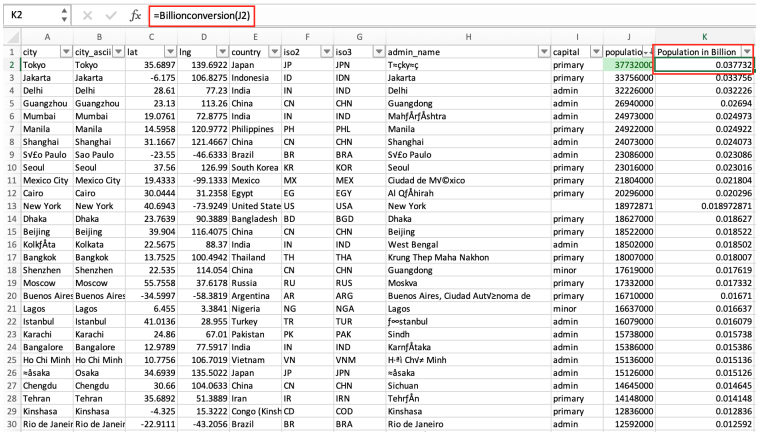


Figure 10.16: Creating a function for converting population in Billion



	A	B	C	D	E	F	G	H	I	J	K
1	city	city_ascii	lat	lng	country	iso2	iso3	admin_name	capital	populatio-	Population in Billion
2	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	T¤ky¤ç	primary	37732000	0.037732
3	Jakarta	Jakarta	-6.175	106.8275	Indonesia	ID	IDN	Jakarta	primary	33756000	0.033756
4	Delhi	Delhi	28.61	77.23	India	IN	IND	Delhi	admin	32226000	0.032226
5	Guangzhou	Guangzhou	23.13	113.26	China	CN	CHN	Guangdong	admin	26940000	0.02694
6	Mumbai	Mumbai	19.0761	72.8775	India	IN	IND	MahƒÁrƒÅshtra	admin	24973000	0.024973
7	Manila	Manila	14.5958	120.9772	Philippines	PH	PHL	Manila	primary	24922000	0.024922
8	Shanghai	Shanghai	31.1667	121.4667	China	CN	CHN	Shanghai	admin	24073000	0.024073
9	SVEo Paulo	Sao Paulo	-23.55	-46.6333	Brazil	BR	BRA	SVEo Paulo	admin	23086000	0.023086
10	Seoul	Seoul	37.56	126.99	South Korea	KR	KOR	Seoul	primary	23016000	0.023016
11	Mexico City	Mexico City	19.4333	-99.1333	Mexico	MX	MEX	Ciudad de MV©xico	primary	21804000	0.021804
12	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	Al QƒÁhirah	primary	20296000	0.020296
13	New York	New York	40.6943	-73.9249	United State	US	USA	New York		18972871	0.018972871
14	Dhaka	Dhaka	23.7639	90.3889	Bangladesh	BD	BGD	Dhaka	primary	18627000	0.018627
15	Beijing	Beijing	39.904	116.4075	China	CN	CHN	Beijing	primary	18522000	0.018522
16	KolkÍÁta	Kolkata	22.5675	88.37	India	IN	IND	West Bengal	admin	18502000	0.018502
17	Bangkok	Bangkok	13.7525	100.4942	Thailand	TH	THA	Krung Thep Maha Nakhon	primary	18007000	0.018007
18	Shenzhen	Shenzhen	22.535	114.054	China	CN	CHN	Guangdong	minor	17619000	0.017619
19	Moscow	Moscow	55.7558	37.6178	Russia	RU	RUS	Moskva	primary	17332000	0.017332
20	Buenos Aires	Buenos Aires	-34.5997	-58.3819	Argentina	AR	ARG	Buenos Aires, Ciudad AutV2noma de	primary	16710000	0.01671
21	Lagos	Lagos	6.455	3.3841	Nigeria	NG	NGA	Lagos	minor	16637000	0.016637
22	Istanbul	Istanbul	41.0136	28.955	Turkey	TR	TUR	ƒ¬estanbul	admin	16079000	0.016079
23	Karachi	Karachi	24.86	67.01	Pakistan	PK	PAK	Sindh	admin	15738000	0.015738
24	Bangalore	Bangalore	12.9789	77.5917	India	IN	IND	KarnƒÁtaka	admin	15386000	0.015386
25	Ho Chi Minh	Ho Chi Minh	10.7756	106.7019	Vietnam	VN	VNM	H·Ì Chv¥ Minh	admin	15136000	0.015136
26	¤åsaka	Osaka	34.6939	135.5022	Japan	JP	JPN	¤åsaka	admin	15126000	0.015126
27	Chengdu	Chengdu	30.66	104.0633	China	CN	CHN	Sichuan	admin	14645000	0.014645
28	Tehran	Tehran	35.6892	51.3889	Iran	IR	IRN	TehrƒÁn	primary	14148000	0.014148
29	Kinshasa	Kinshasa	-4.325	15.3222	Congo (Kinsh	CD	COD	Kinshasa	primary	12836000	0.012836
30	Rio de Janeir	Rio de Janeir	-22.9111	-43.2056	Brazil	BR	BRA	Rio de Janeiro	admin	12592000	0.012592

Figure 10.17: Calling the user defined functionality to convert population in Billion

4. Conclude the UDF definition by entering the End Function command on the last line.