TOLANI COLLEGE OF COMMERCE
(Affiliated to University of Mumbai)
Sher E Punjab Colony, Andheri East
MUMBAI-MAHARASHTRA-400093
DEPARTMENT OF BSc (INFORMATION TECHNOLOGY)

CERTIFICATE

This is to certify that the Journal entitled, "**Business Intelligence**", is bonafied work of **Yuvraj Vijay Achrekar** bearing roll no: **03** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai, during the academic year 2022 - 2023.

Internal Examiner **Coordinator**

External Examiner

Date: **College Seal**

INDEX

Sr. No.	Date	Practical	Sign
0		Installation and Setup of Softwares A) Power BI B) R Studio	
1		Import the Legacy Data from Different Sources A) Importing Excel File B) Importing O-Data File	
2		ETL Process A) Removing columns B) Changing Data Types C) Expanding the Order Details table D) Calculating Line Total E) Renaming and Reordering Columns F) Combine the Products and Total Sales	
3		Implementation of Classification Algorithm inR Programming	
4		K- means Clustering using R Programming	
5		Prediction using Linear Regression in R Programming	
6		Perform the logistic regression on the given data warehouse data	
7		Apply What-if Analysis for DataVisualization.	
8		Data Analyzing using Time-series Analysis.	
9		Implementation of Decision Tree using R Tool	

PRACTICAL 0 – Installation and Software Setup

A.) Power Bi Software
B.) R Studio Software
C.) SQL Server 2012 Software Setup and Installation:

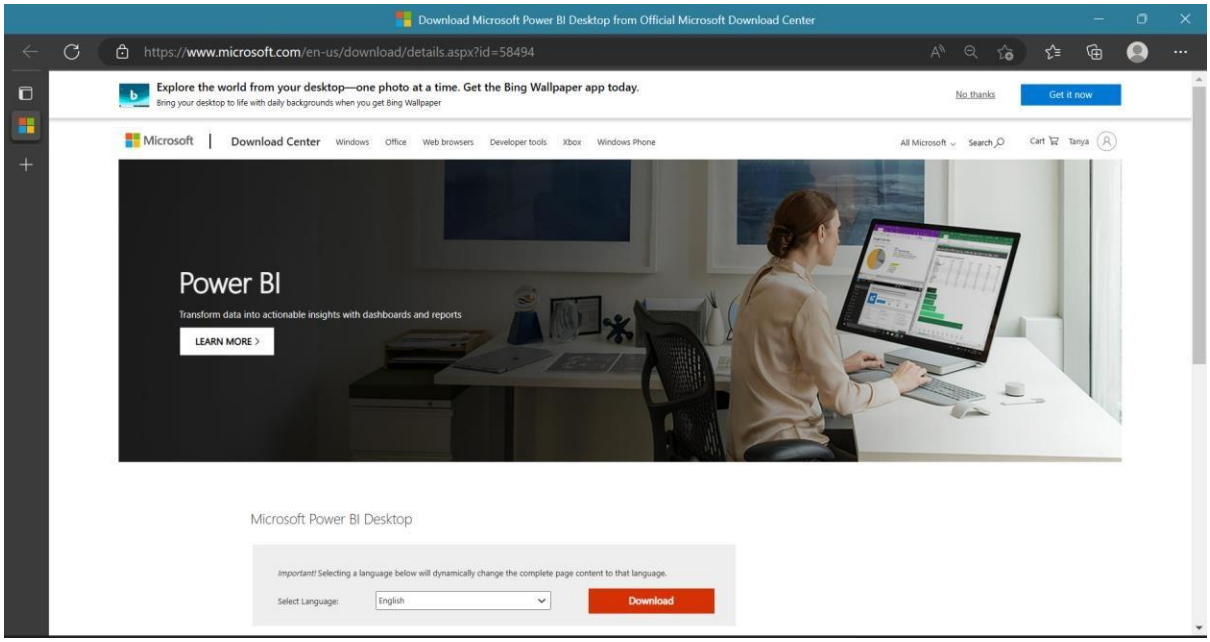
A) Power Bi Software Setup and Installation:

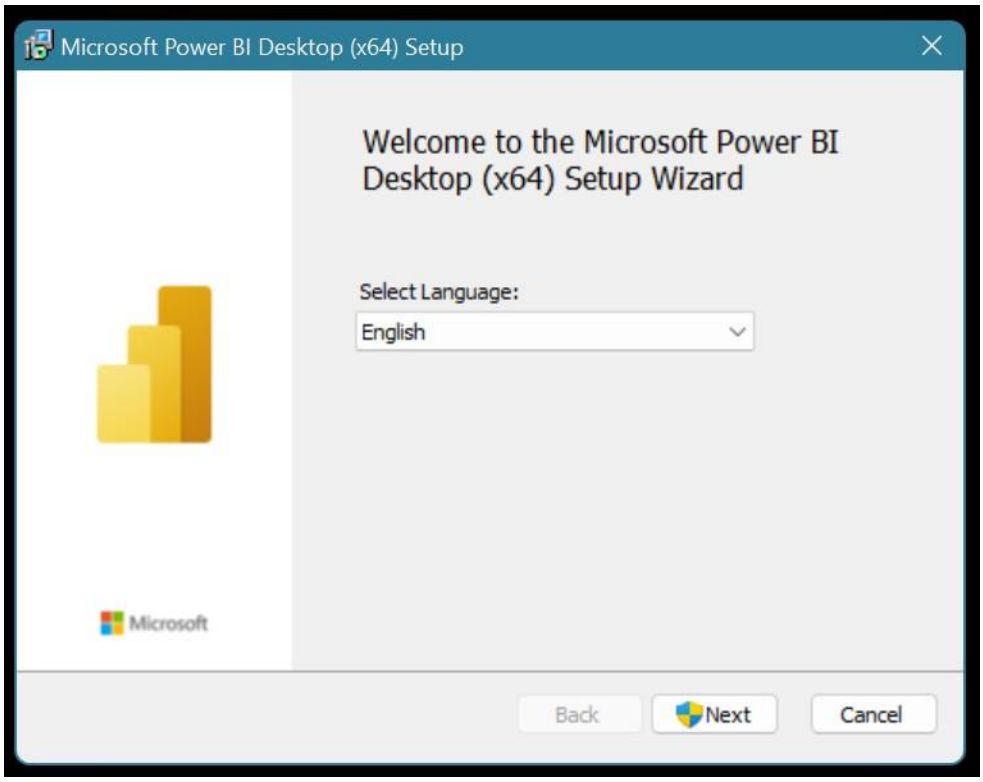
Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive and interactiveinsights. Your data may be an Excel spreadsheet, or a collection of cloud-based and on- premises hybrid data warehouses, Power BI lets you easily connect to your data sources, visualize, and discover what's important, and share that with anyone or everyone you want.

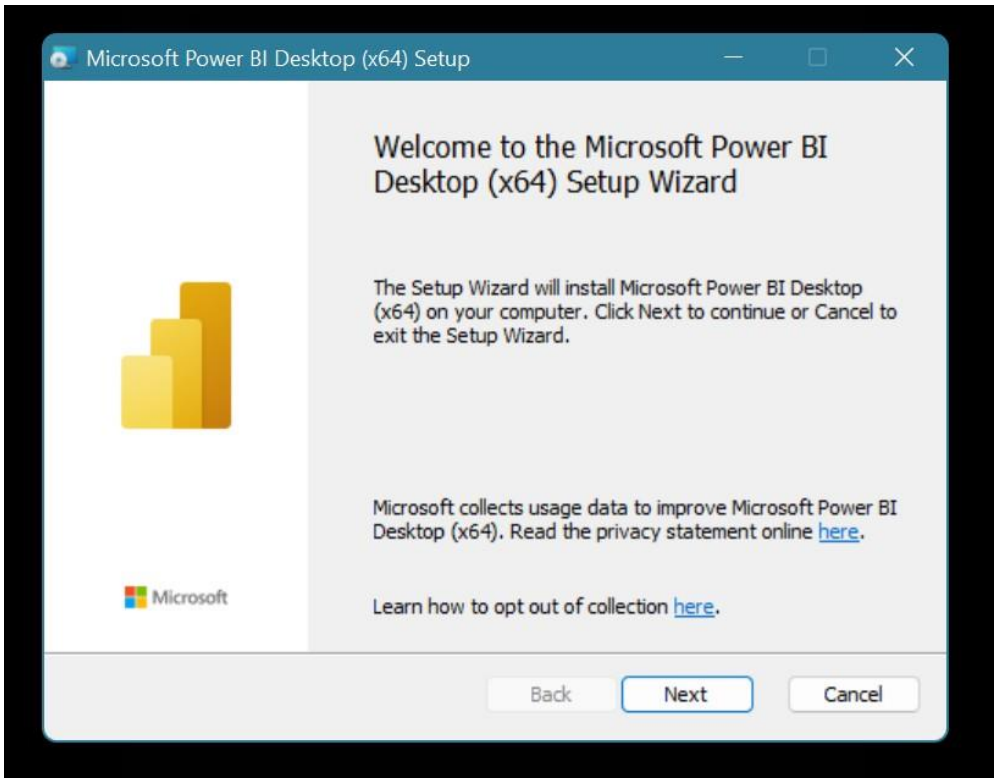
Download Power Bi software by clicking on the link given below and follow the following screenshots for further setup steps:-

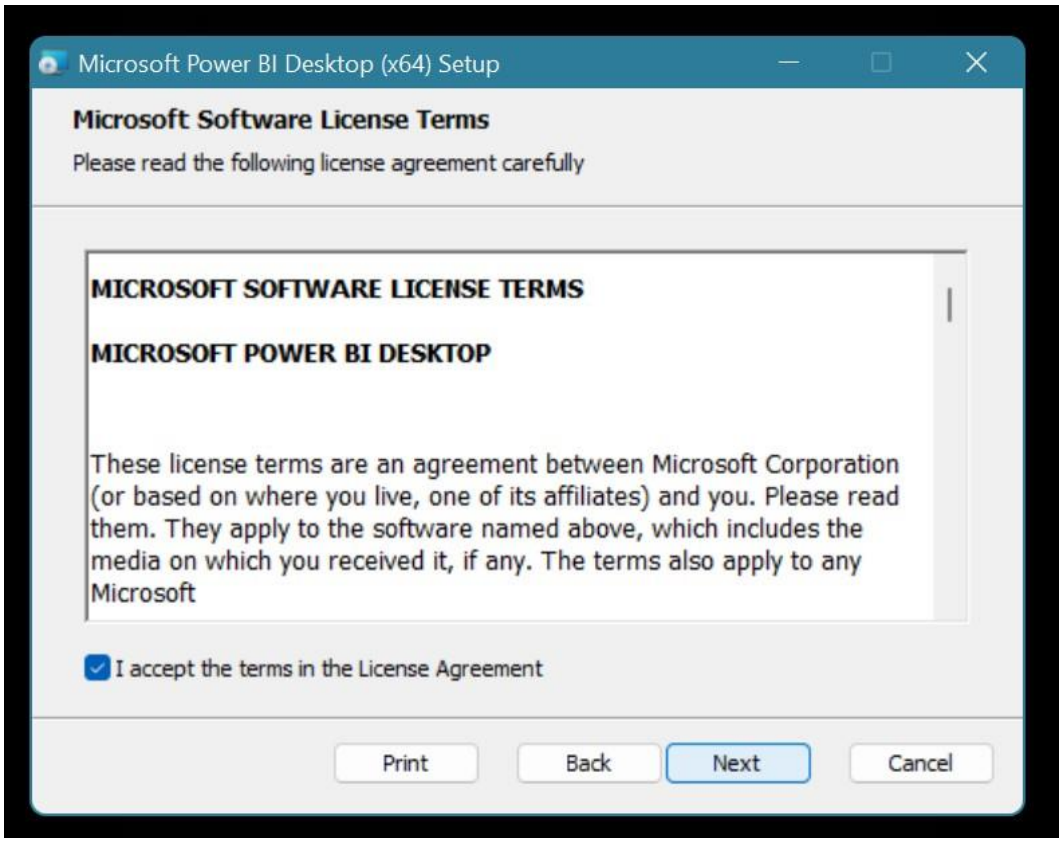
Download Microsoft Power BI Desktop from Official Microsoft Download Center

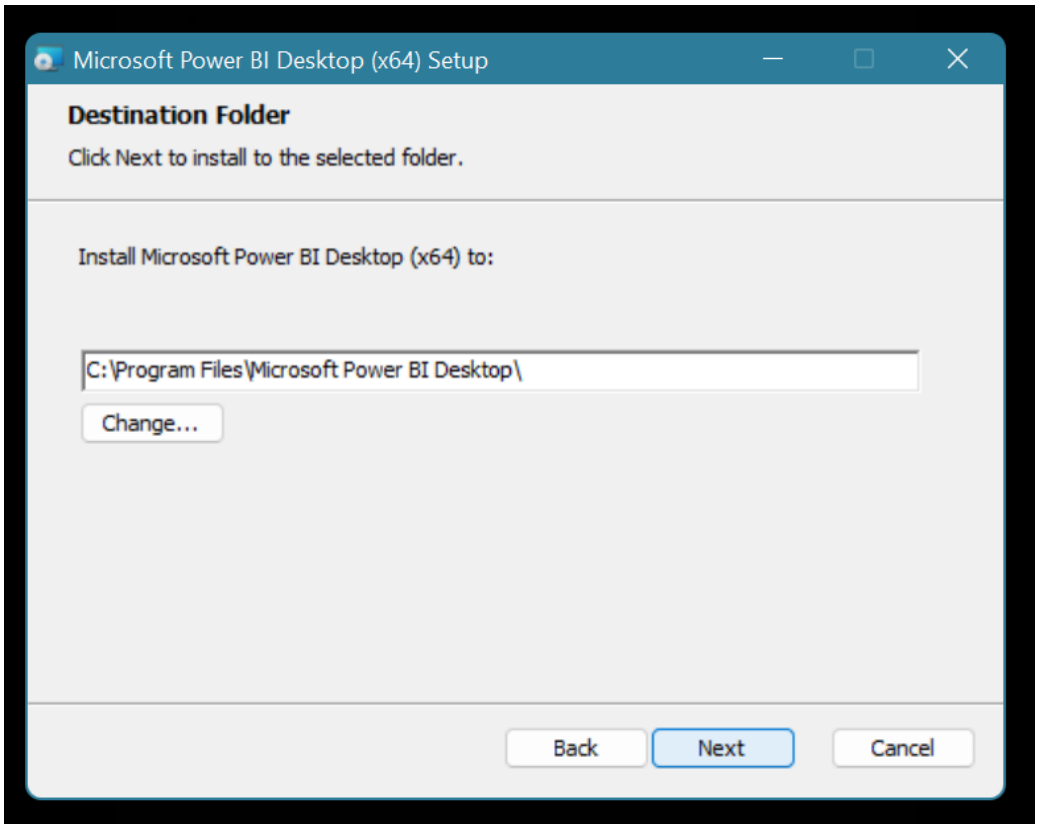
https://www.microsoft.com/en-us/download/details.aspx?id=58494

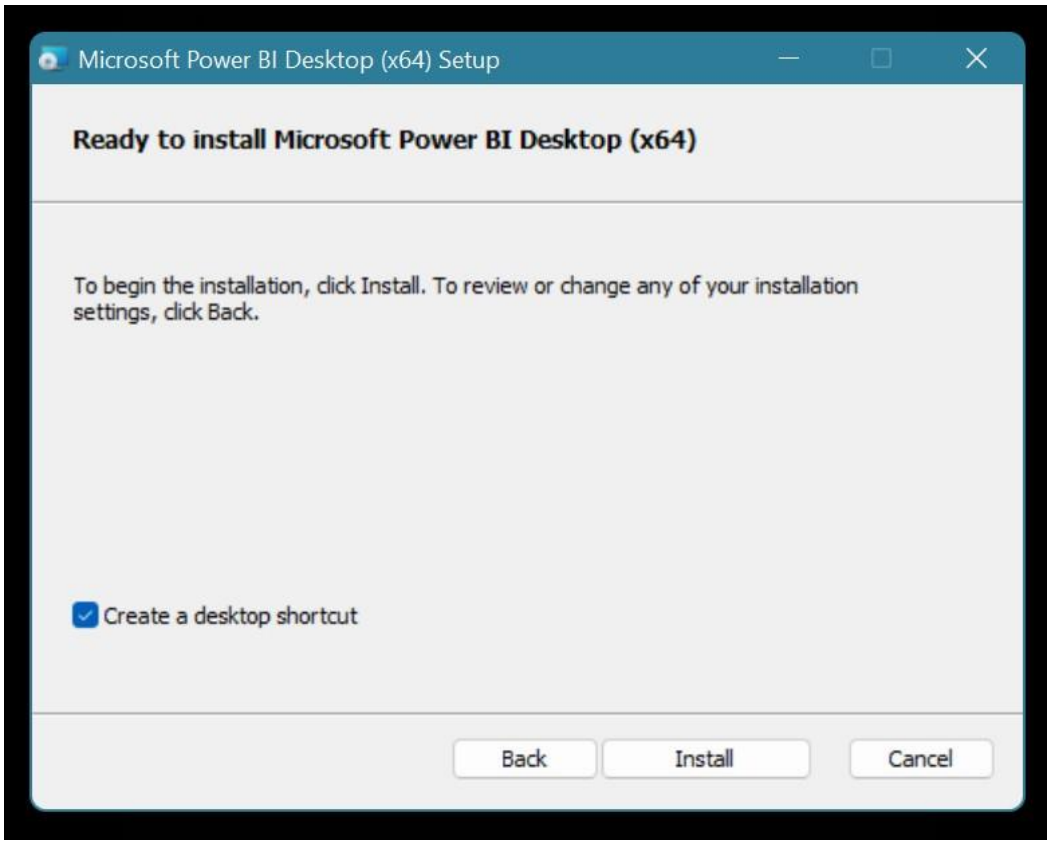


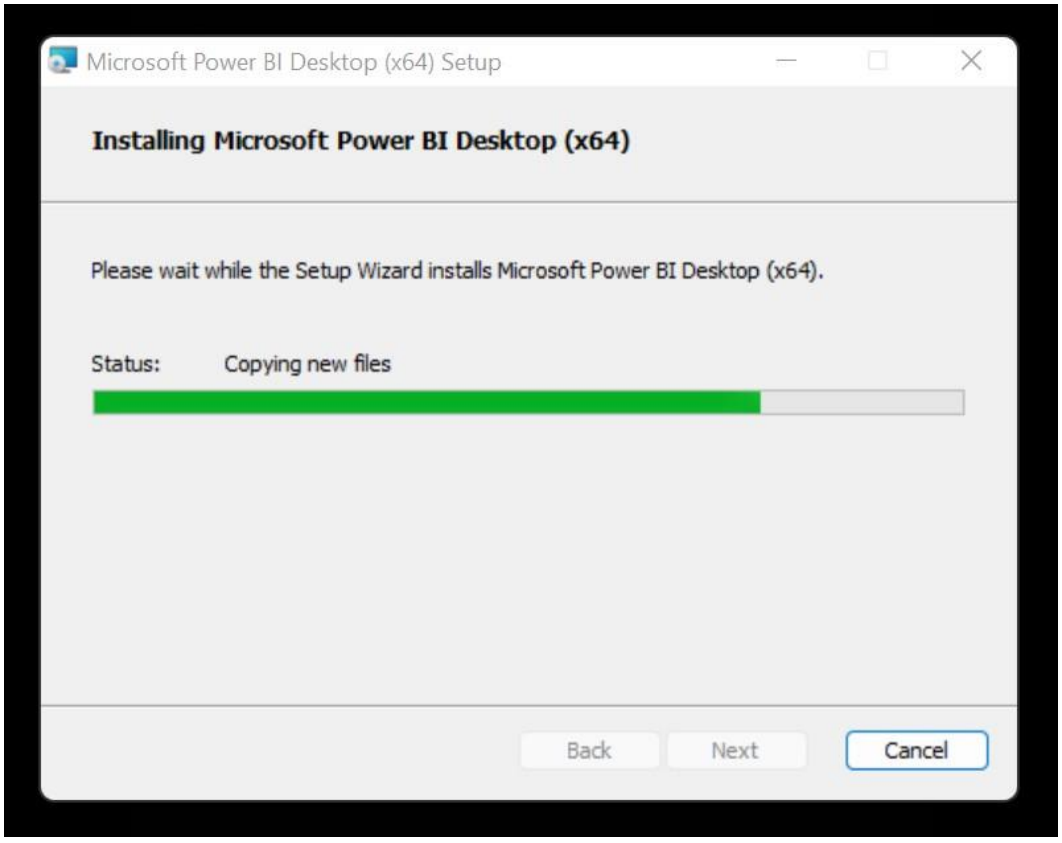

Microsoft Power BI Desktop (x64) Setup
Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard
Select Language:
English
Microsoft
Back
Next
Cancel

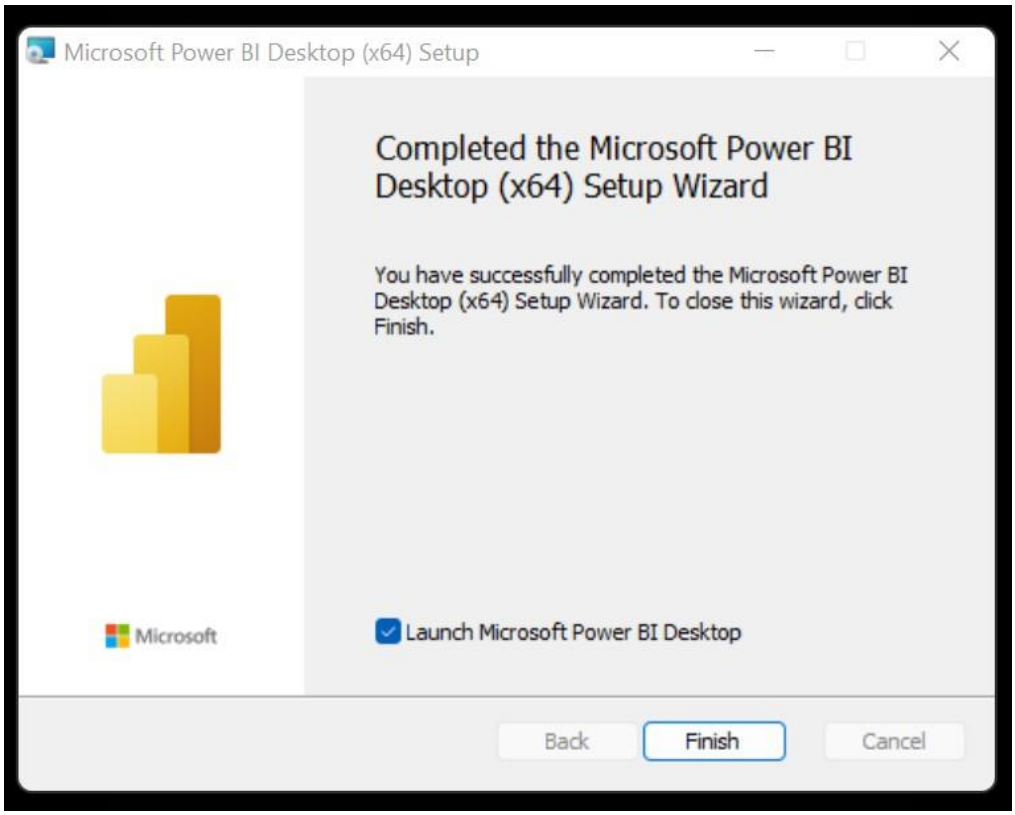

Microsoft Power BI Desktop (x64) Setup
Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard
The Setup Wizard will install Microsoft Power BI Desktop (x64) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Microsoft collects usage data to improve Microsoft Power BI Desktop (x64). Read the privacy statement online here.
Learn how to opt out of collection here.
Microsoft
Back
Next
Cancel


Microsoft Power BI Desktop (x64) Setup
Microsoft Software License Terms
Please read the following license agreement carefully
MICROSOFT SOFTWARE LICENSE TERMS
MICROSOFT POWER BI DESKTOP
These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft
I accept the terms in the License Agreement
Print
Back
Next
Cancel


Microsoft Power BI Desktop (x64) Setup
Destination Folder
Click Next to install to the selected folder.
Install Microsoft Power BI Desktop (x64) to:
C:\Program Files\Microsoft Power BI Desktop\
Change...
Back
Next
Cancel


Microsoft Power BI Desktop (x64) Setup
Ready to install Microsoft Power BI Desktop (x64)
To begin the installation, click Install. To review or change any of your installation settings, click Back.
Create a desktop shortcut
Back
Install
Cancel

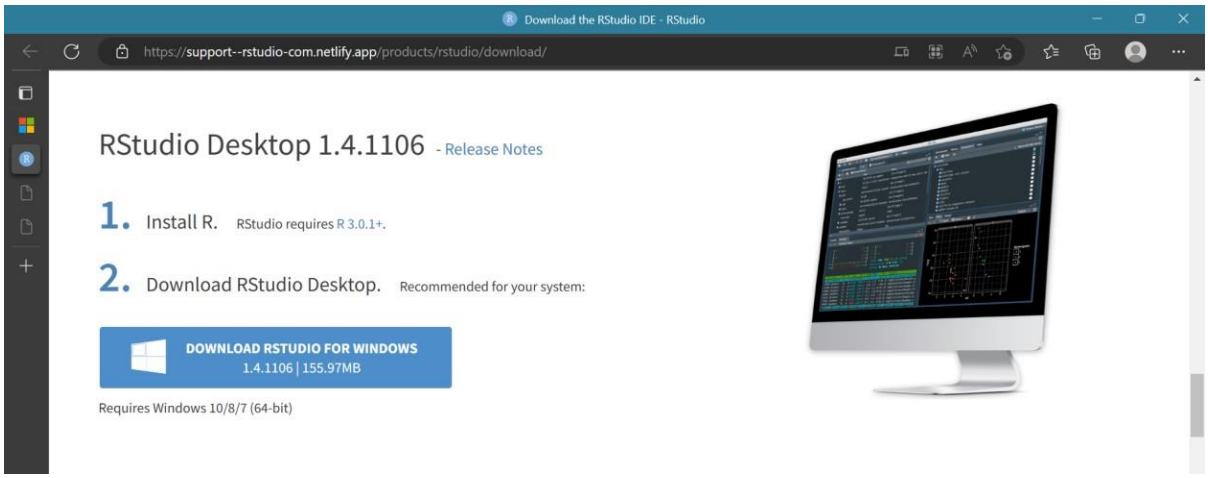

Microsoft Power BI Desktop (x64) Setup
Installing Microsoft Power BI Desktop (x64)
Please wait while the Setup Wizard installs Microsoft Power BI Desktop (x64).
Status: Copying new files
Back
Next
Cancel


Microsoft Power BI Desktop (x64) Setup
Completed the Microsoft Power BI Desktop (x64) Setup Wizard
You have successfully completed the Microsoft Power BI Desktop (x64) Setup Wizard. To close this wizard, click Finish.
Launch Microsoft Power BI Desktop
Microsoft
Back
Finish
Cancel

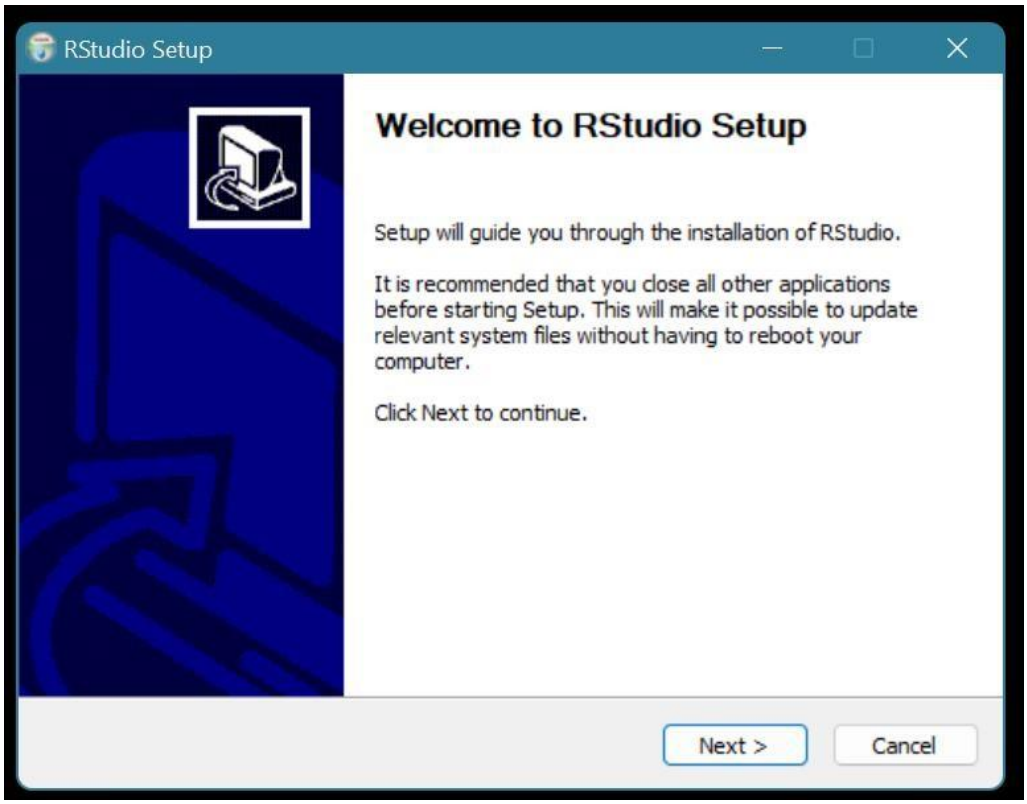
B) R Studio Software Setup and Installation:

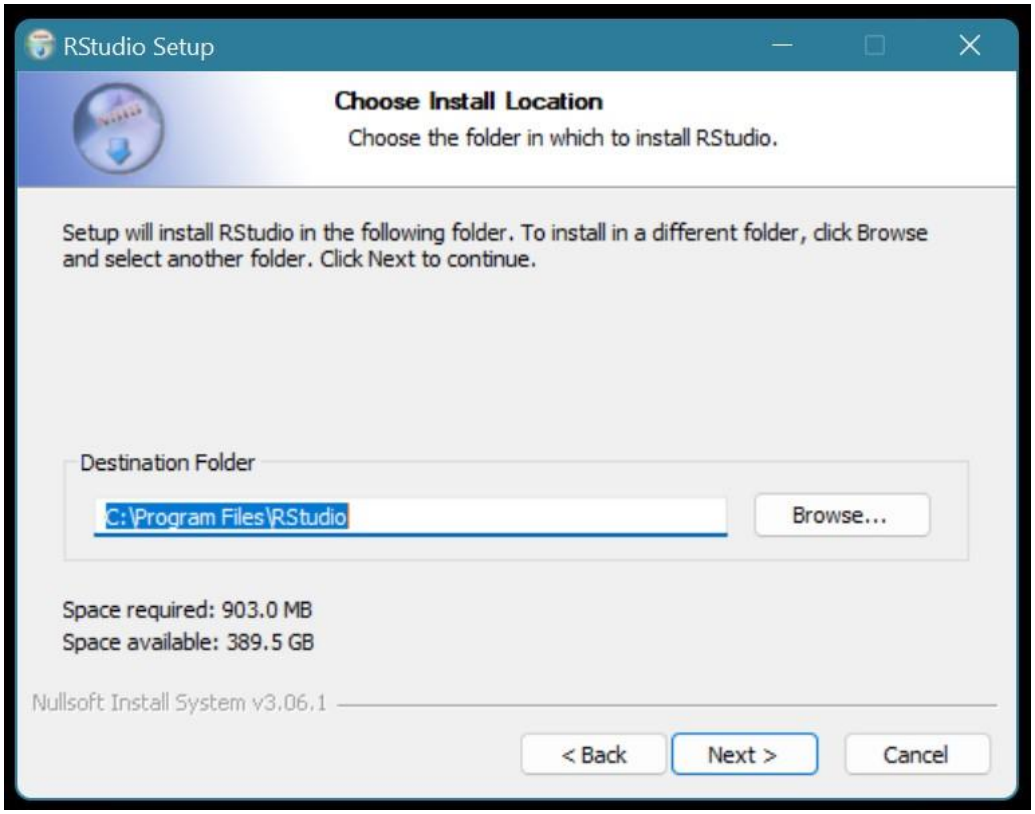
Download R Studio software by clicking on the link given below and follow the following screenshots for further setup steps:-

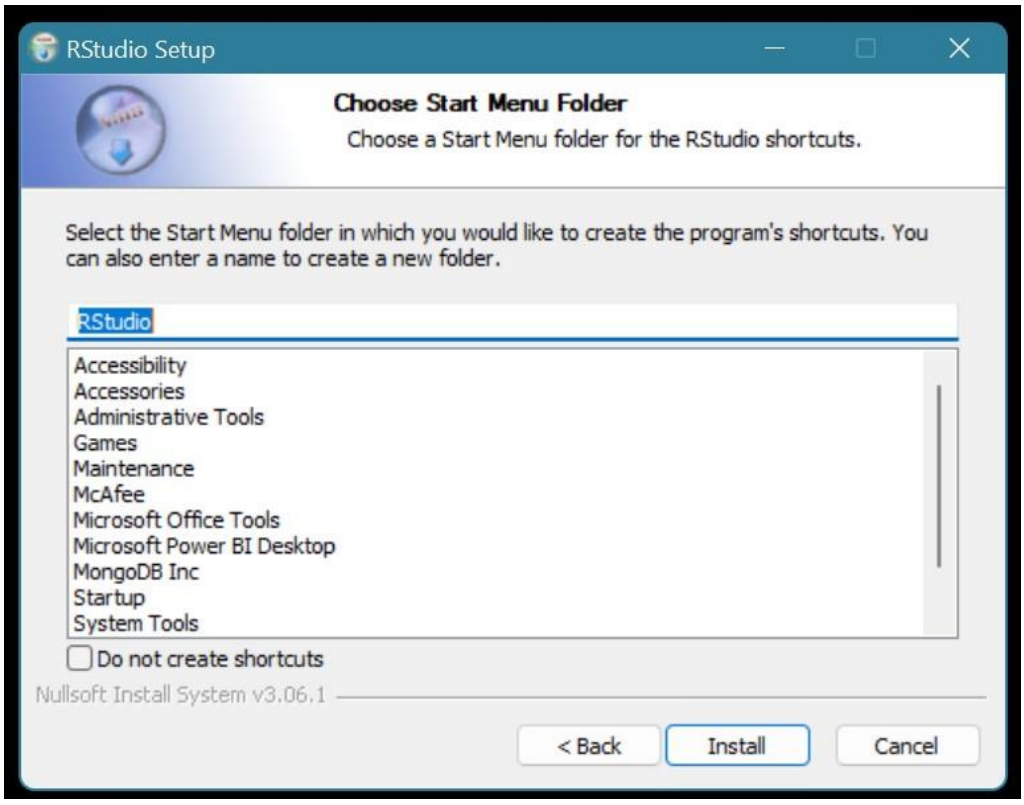
[Download the RStudio IDE - RStudio (support--rstudio-com.netlify.app)]

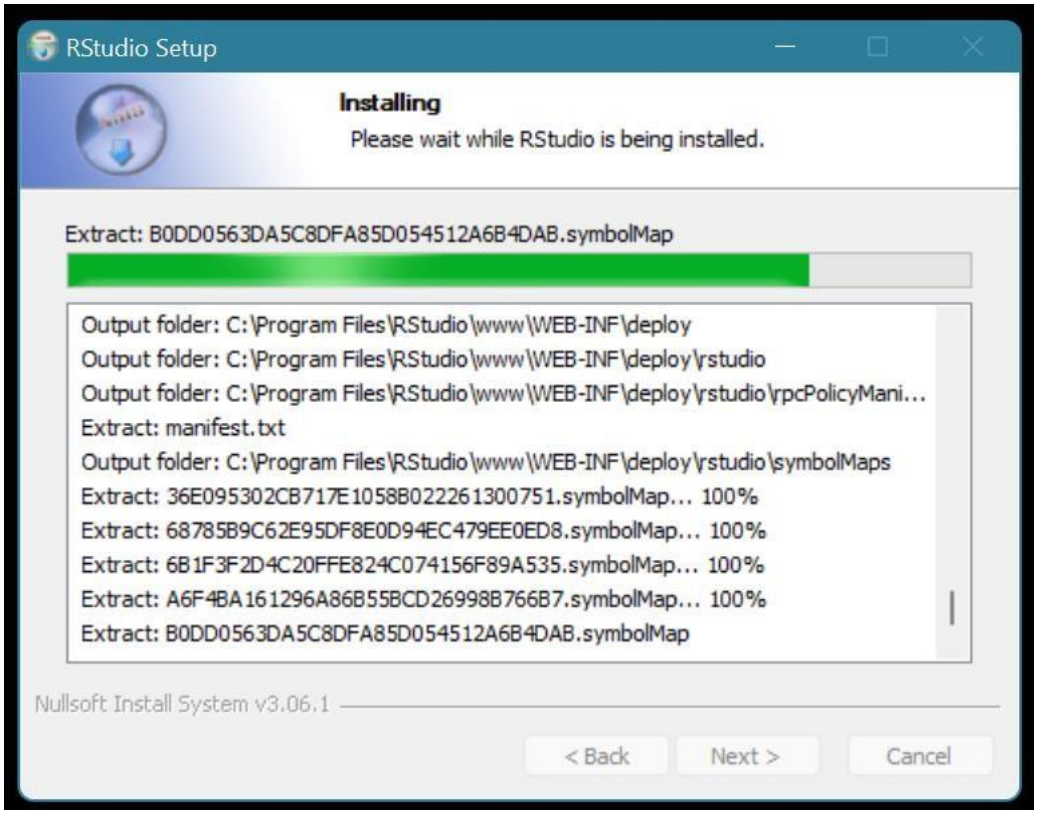


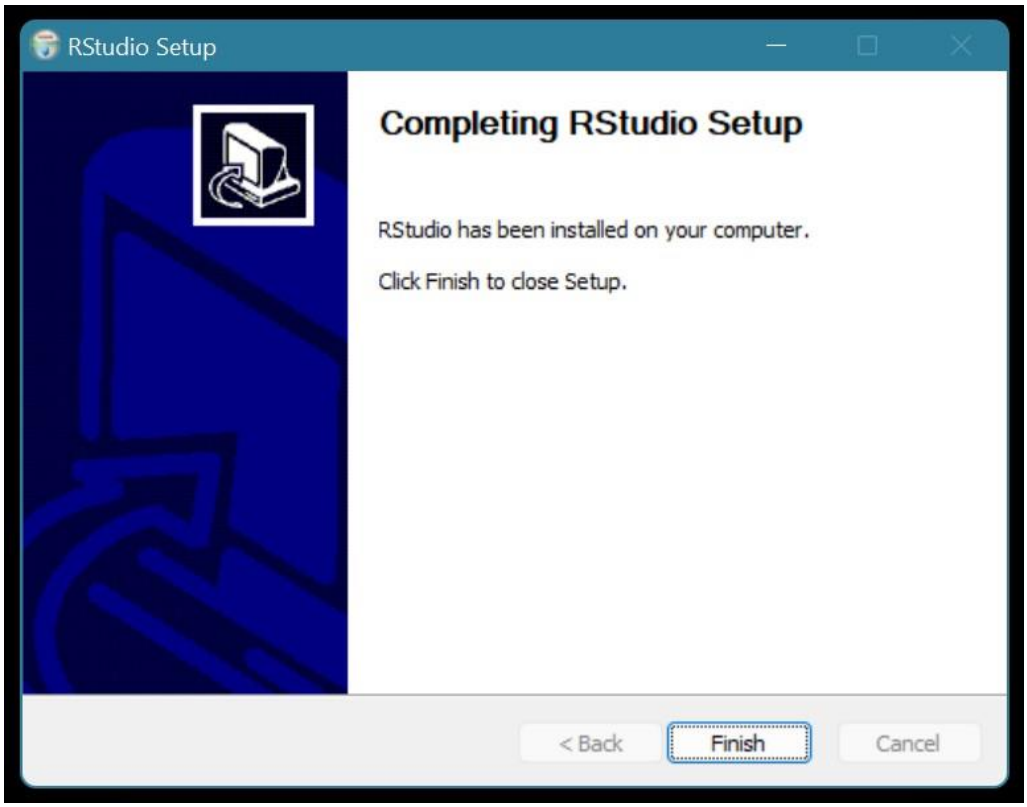
Download




RStudio Setup
Choose Install Location
Choose the folder in which to install RStudio.
Setup will install RStudio in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Program Files\RStudio
Browse...
Space required: 903.0 MB
Space available: 389.5 GB
Nullsoft Install System v3.06.1
< Back
Next >
Cancel


RStudio Setup
Choose Start Menu Folder
Choose a Start Menu folder for the RStudio shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
RStudio
Accessibility
Accessories
Administrative Tools
Games
Maintenance
McAfee
Microsoft Office Tools
Microsoft Power BI Desktop
MongoDB Inc
Startup
System Tools
Do not create shortcuts
Nullsoft Install System v3.06.1
< Back
Install
Cancel


RStudio Setup
Installing
Please wait while RStudio is being installed.
Extract: B0DD0563DA5C8DFA85D054512A6B4DAB.symbolMap
Output folder: C:\Program Files\RStudio\www\WEB-INF\deploy
Output folder: C:\Program Files\RStudio\www\WEB-INF\deploy\rstudio
Output folder: C:\Program Files\RStudio\www\WEB-INF\deploy\rstudio\rpcPolicyMani...
Extract: manifest.txt
Output folder: C:\Program Files\RStudio\www\WEB-INF\deploy\rstudio\symbolMaps
Extract: 36E095302CB717E1058B022261300751.symbolMap... 100%
Extract: 68785B9C62E95DF8E0D94EC479EE0ED8.symbolMap... 100%
Extract: 6B1F3F2D4C20FFE824C074156F89A535.symbolMap... 100%
Extract: A6F4BA161296A86B55BCD26998B766B7.symbolMap... 100%
Extract: B0DD0563DA5C8DFA85D054512A6B4DAB.symbolMap
Nullsoft Install System v3.06.1
< Back
Next >
Cancel


RStudio Setup
Completing RStudio Setup
RStudio has been installed on your computer.
Click Finish to close Setup.
< Back
Finish
Cancel

Practical 1

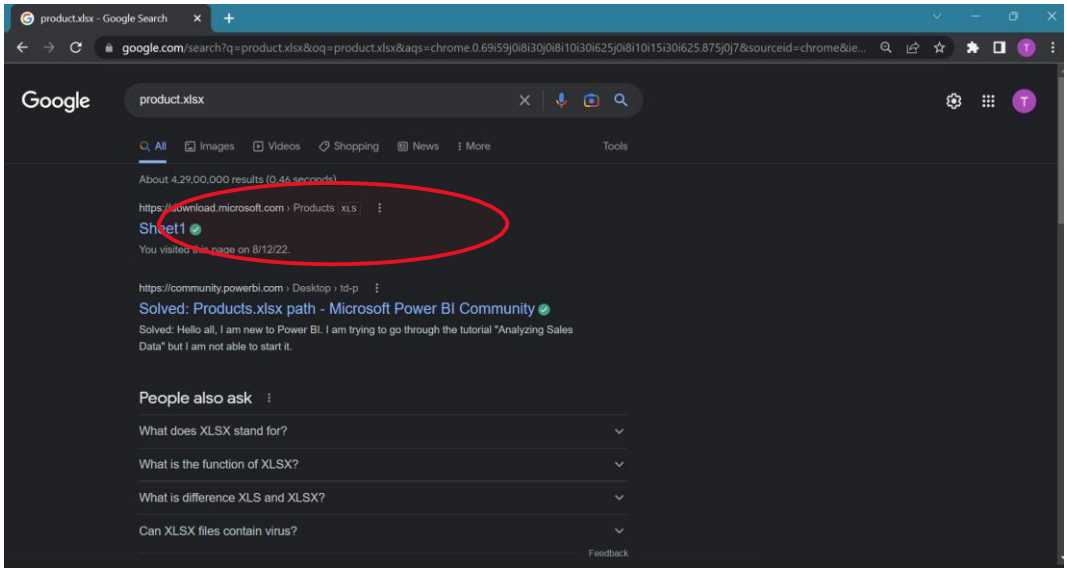
Import the legacy data from different sources such as (Excel, SQL Server, Oracle etc.) and load in the target system.

A) Importing Excel File :–

Step 1: Downloading excel file

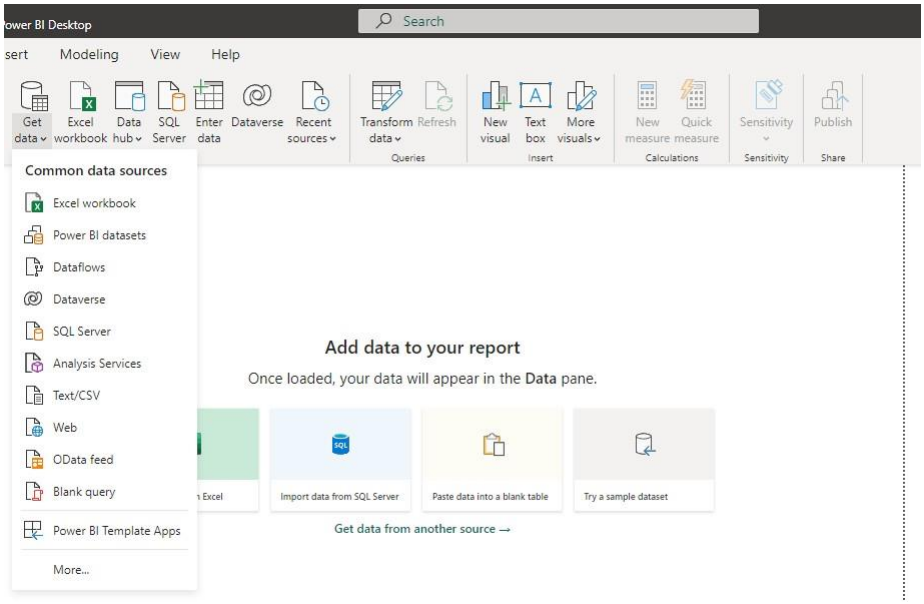
- To download excel file click on the link - https://www.google.com/search?q=product.xlsx&oq=product.xlsx&aqs=chrome..69i57j0i8i30j0i8i10i30i625j0i8i10i15i30i625.767j0j7&sourceid=chrome&ie=UTF-8

- Click on Sheet 1 to download it.

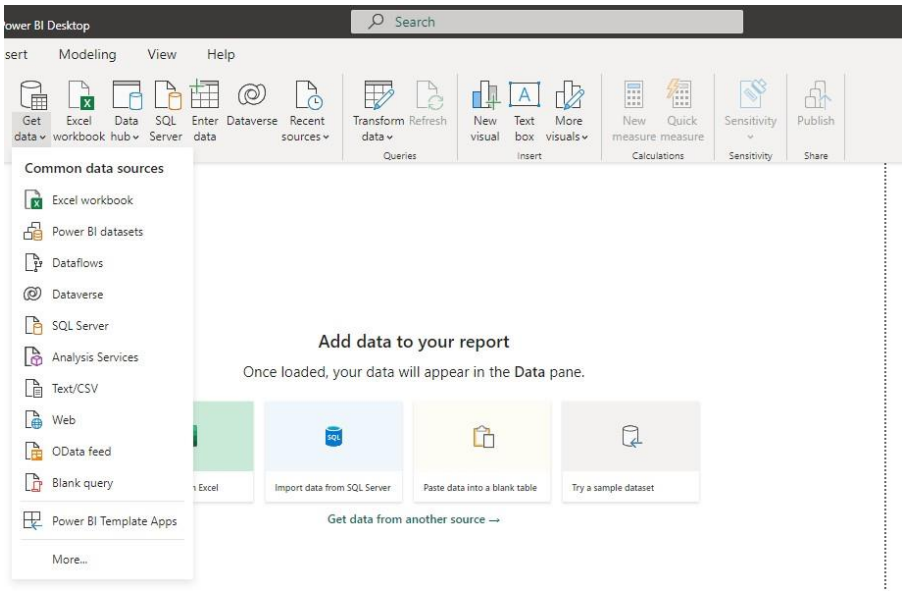


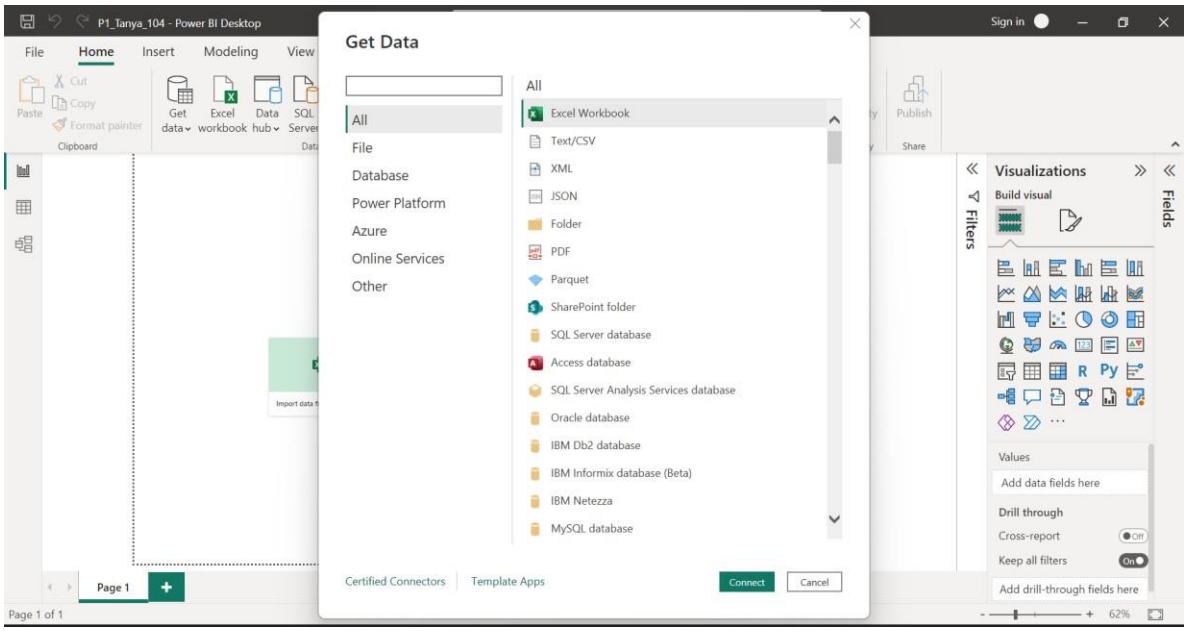
Step 2: Importing Excel file

- To import file, we can directly click on the "import data from excel" option.

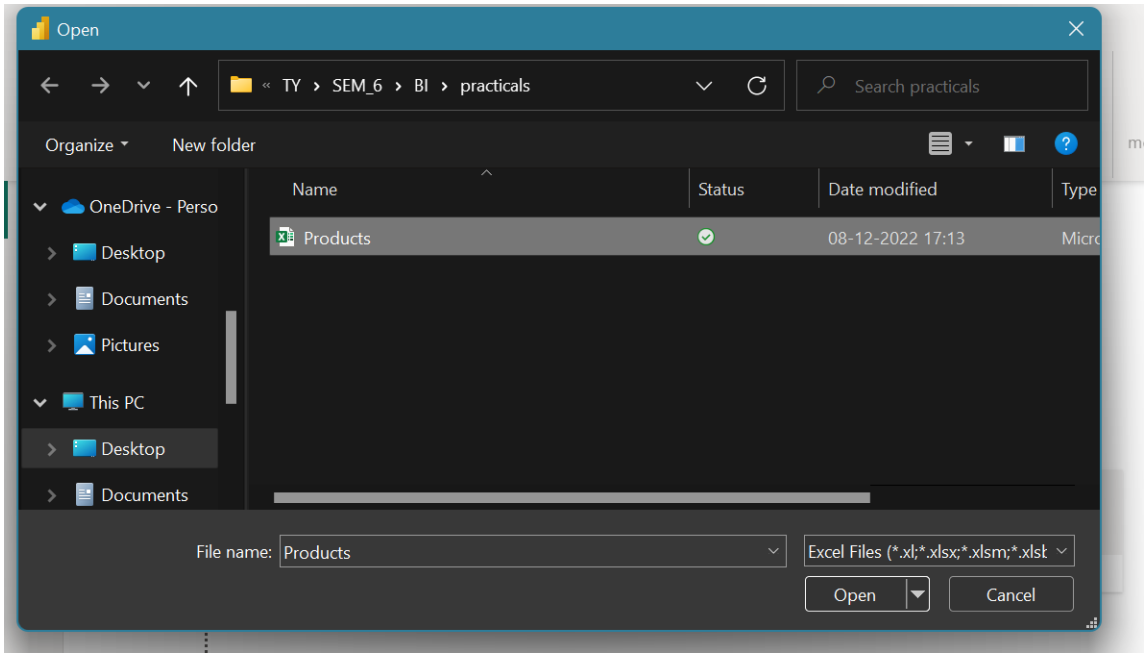


- Also, data can be imported by selecting – "Get data" from "HOME" ribbon : Go to "get data" → select "Excel Workbook" and click on "connect".

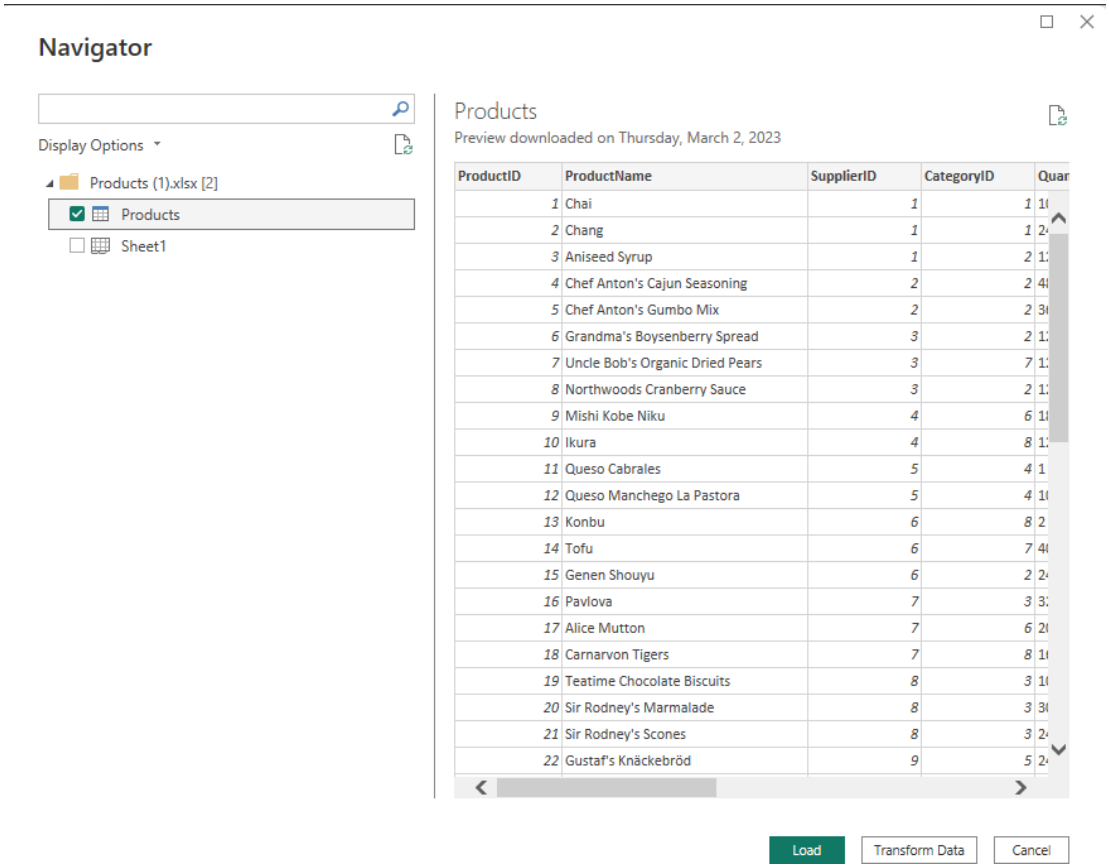




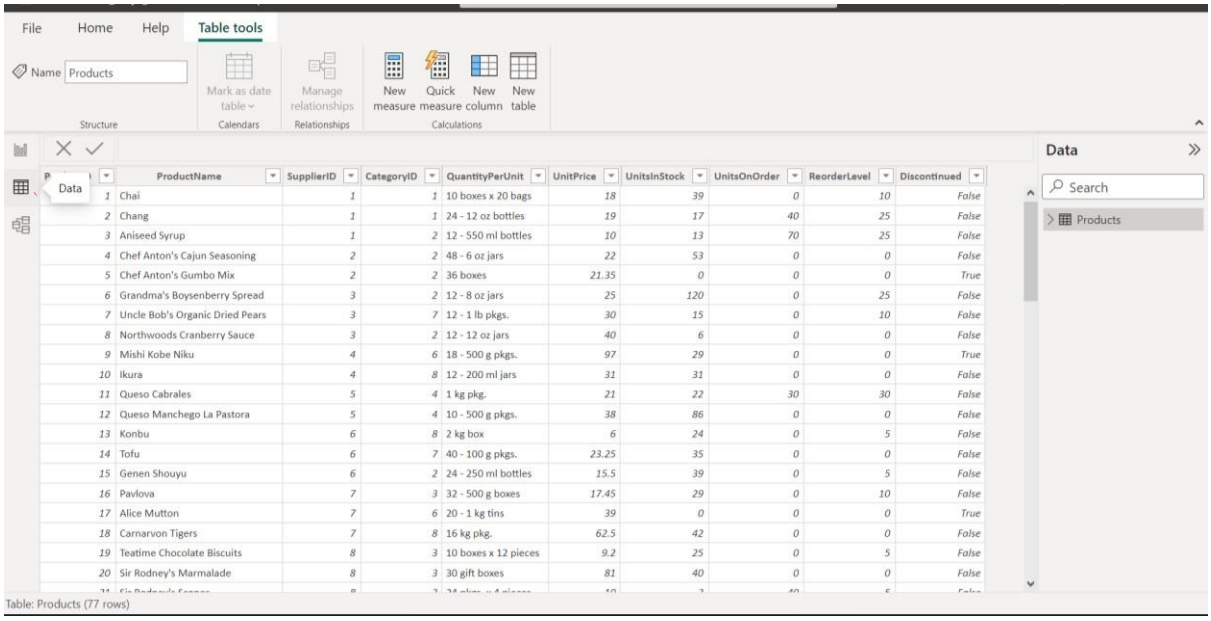
- Select the downloaded file and click on “open”



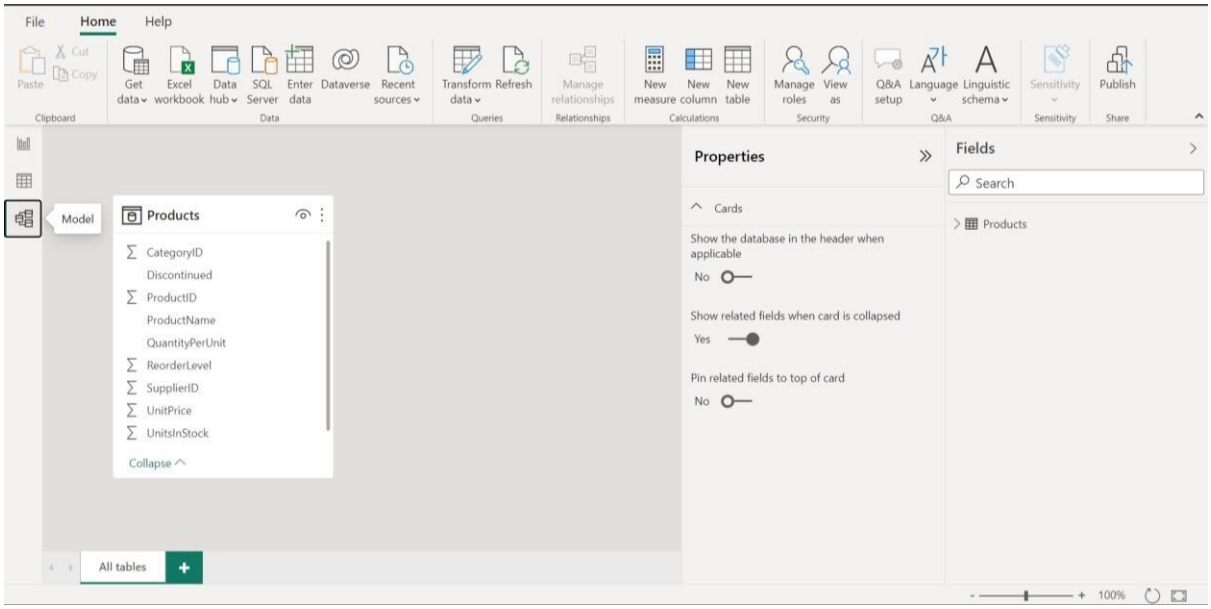
- Select “products” and click on “load”



- From left side, select “data” to display data in tabular format.



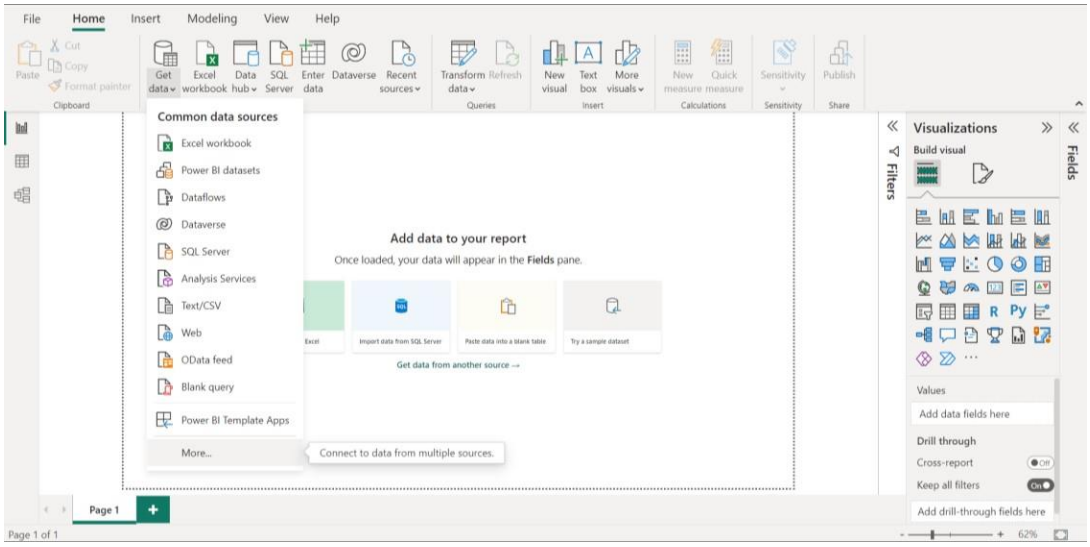
- Similarly, select “Model” to display in a mathematical format.

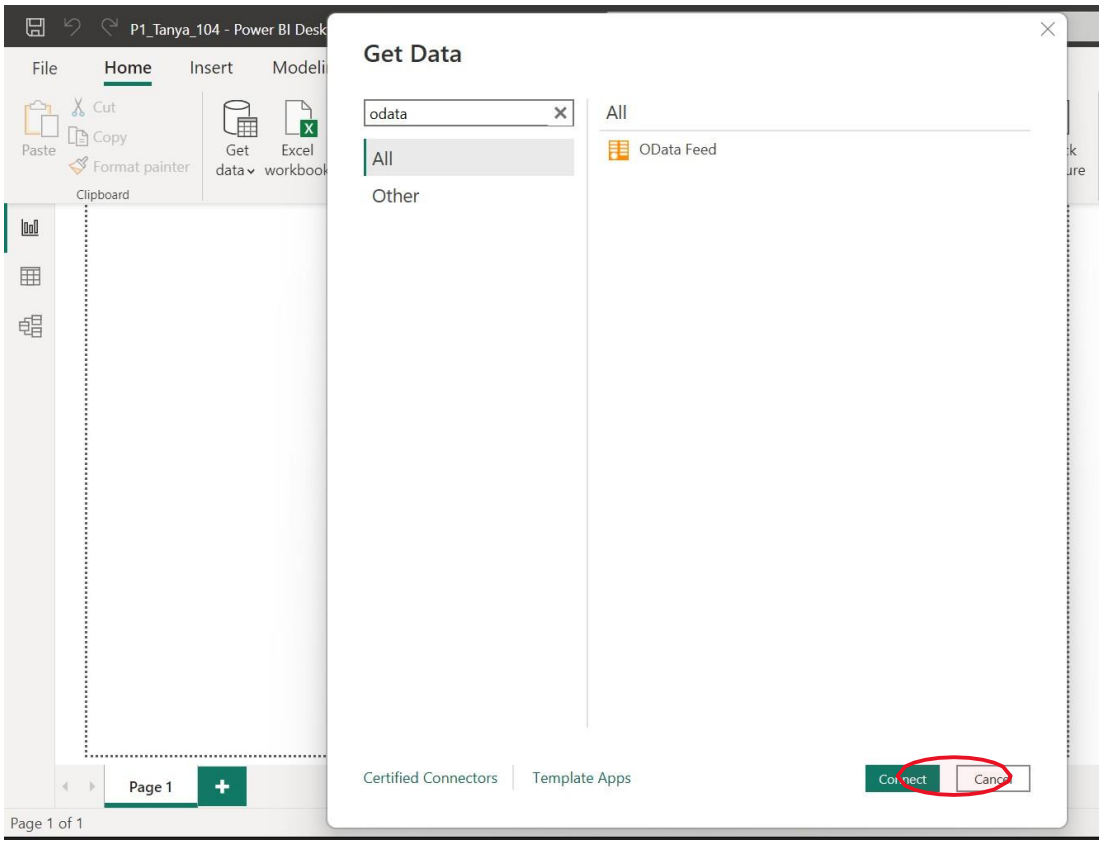


B) Importing O-data File :–

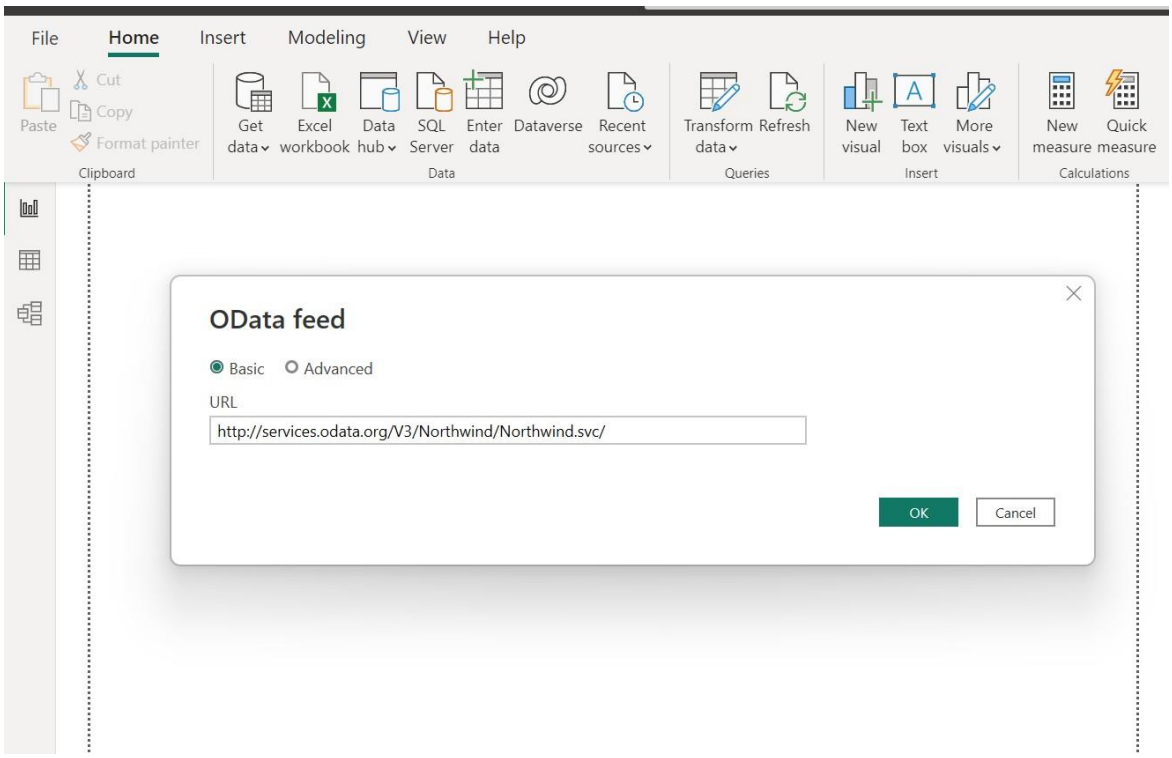
Step 1: Importing O-data file: O-data file is the file that has large amount of data. It is a type of big data.

- Data can be imported by selecting – get data from "HOME" ribbon :
  Go to "get data" → select "OData feed" and click on "connect".

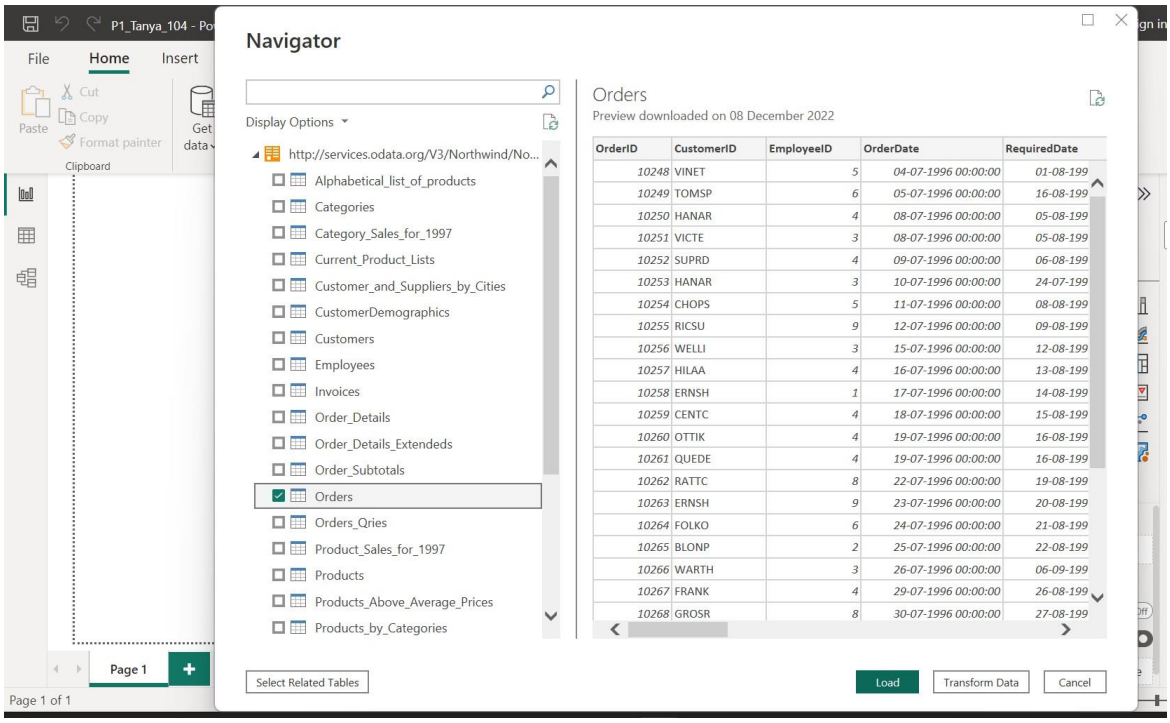




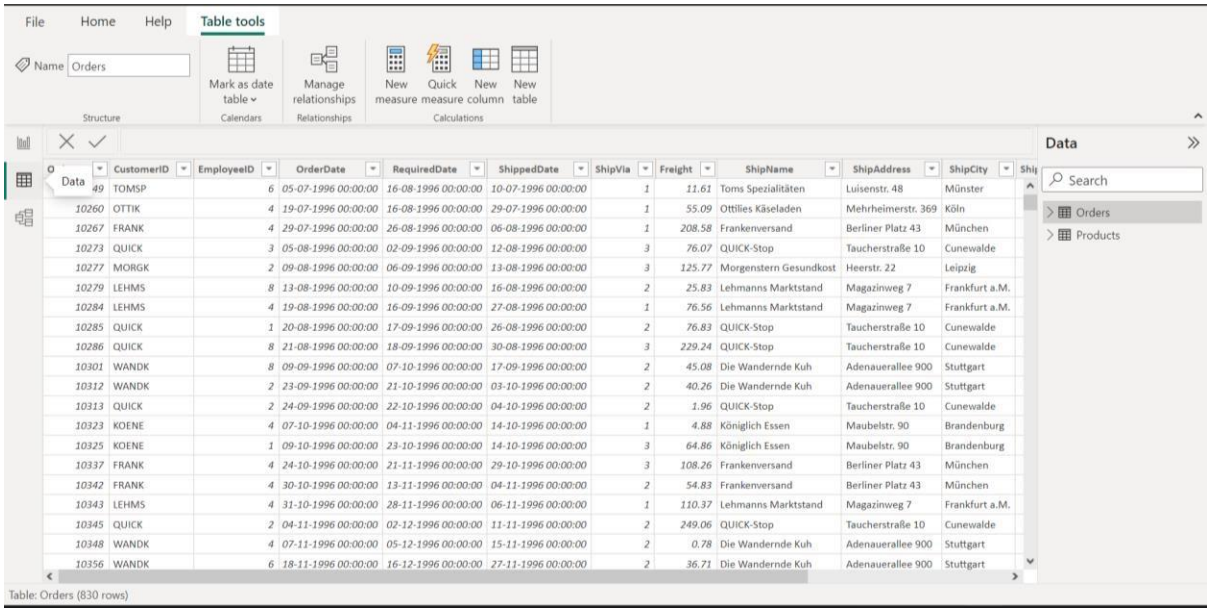
- Now, enter the following URL "http://services.odata.org/V3/Northwind/Northwind.svc/" and click on "OK"



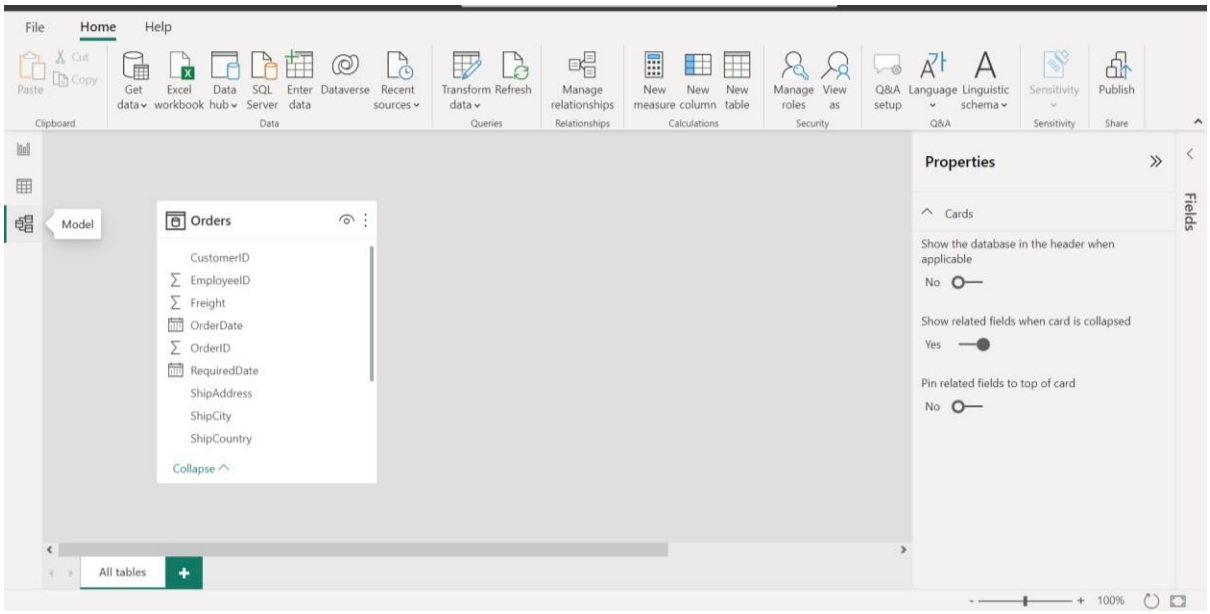
- Select "Orders" and click on "Load"



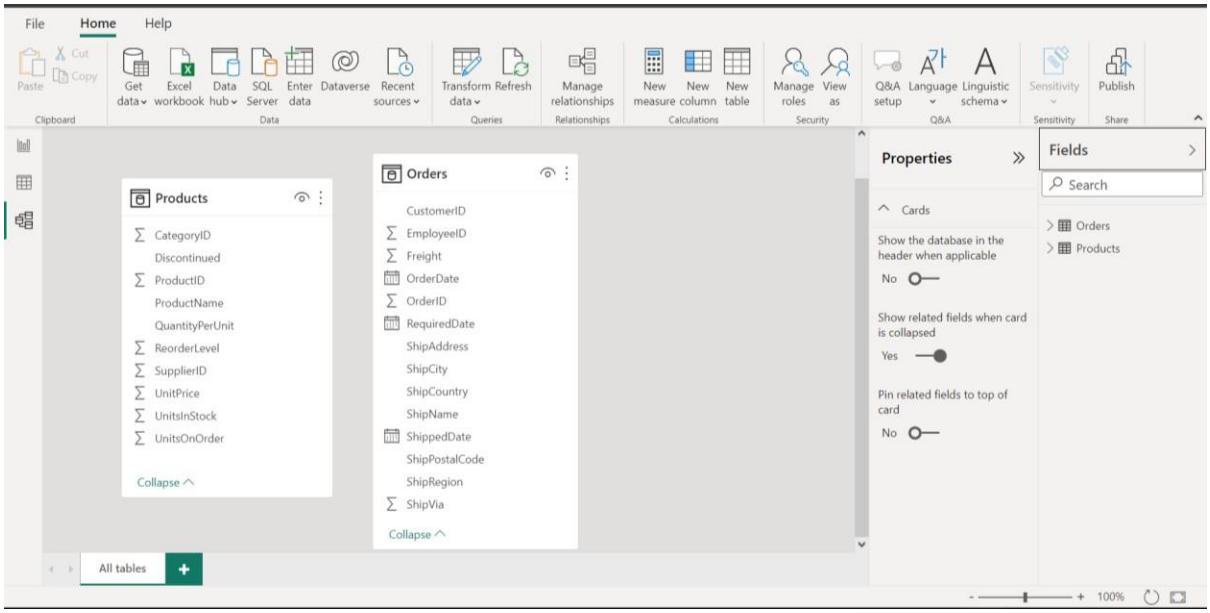
- From left side, select “data” to display data in tabular format.



- Similarly, select “Model” to display in a mathematical format.



- Since, we have loaded the data under same files, we can view it simultaneously.



Practical 2

ETL Process

A) Removing Columns
B) Changing Data types
C) Expanding the Order_Details table
D) Calculating line total
E) Renaming and reordering columns
F) Combine the Products and Total Sales

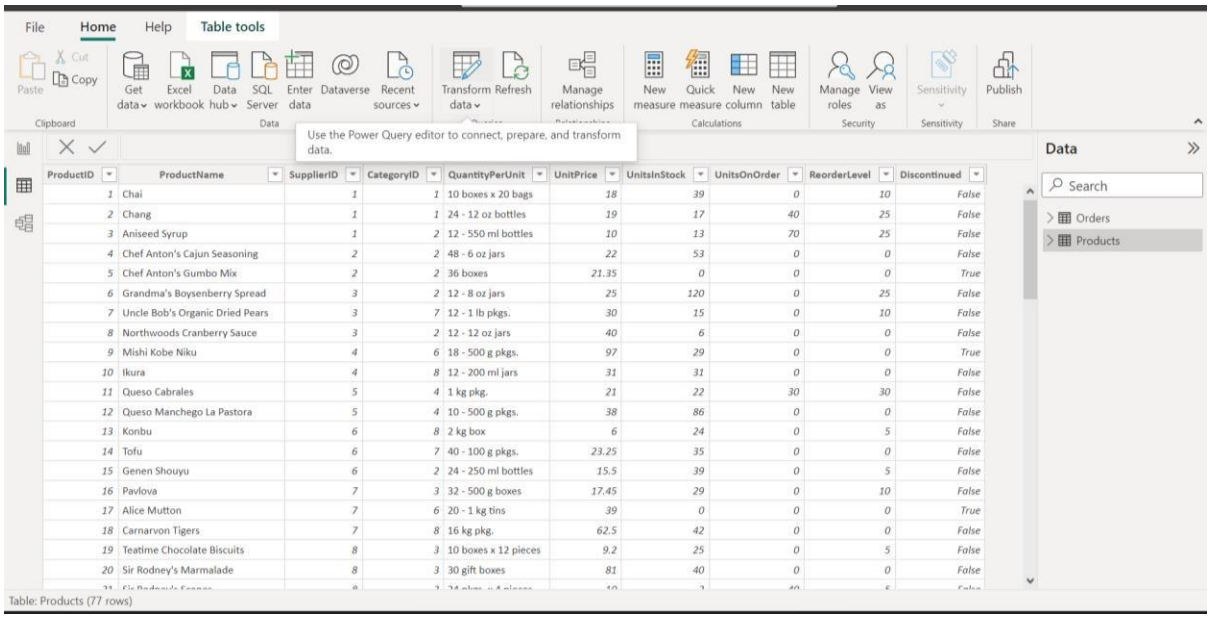
A) Removing other columns to only display columns of interest:–

In this step, we will remove all columns except ProductID, ProductName, UnitsInStock and QuantityPerUnit.

Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select Edit from Navigator. You can also open the Query Editor by selecting Edit Queries from the Home ribbon in Power BI Desktop.

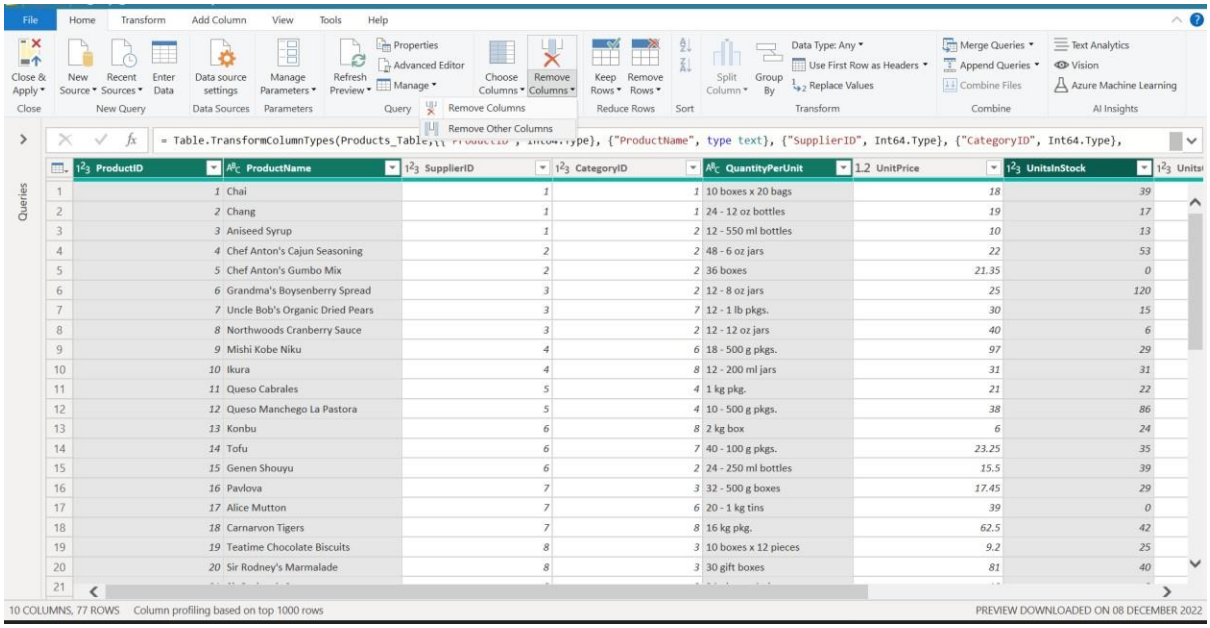
Step 1:

- Select “Transform data” from “Table tools” ribbon.



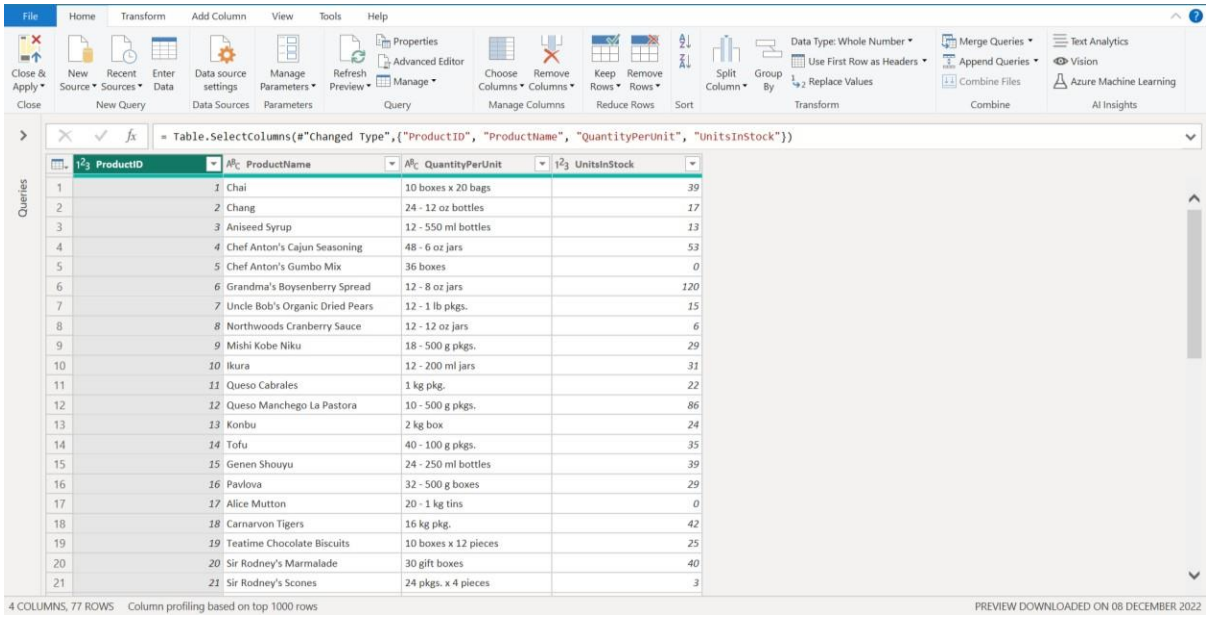
Step 2:

- Press “CTRL” key and select the columns of interest (ProductID, ProductName, QuantityPerUnit, UnitInStock)
- From the “Home” ribbon → Select “Remove Columns” → Click on “Remove Other Columns”



	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	1	Chai	1	1	10 boxes x 20 bags	18	39
2	2	Chang	1	1	24 - 12 oz bottles	19	17
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10	13
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22	53
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	0
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25	120
7	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30	15
8	8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40	6
9	9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97	29
10	10	Ikura	4	8	12 - 200 ml jars	31	31
11	11	Queso Cabrales	5	4	1 kg pkg.	21	22
12	12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38	86
13	13	Konbu	6	8	2 kg box	6	24
14	14	Tofu	6	7	40 - 100 g pkgs.	23.25	35
15	15	Genen Shouyu	6	2	24 - 250 ml bottles	15.5	39
16	16	Pavlova	7	3	32 - 500 g boxes	17.45	29
17	17	Alice Mutton	7	6	20 - 1 kg tins	39	0
18	18	Carnarvon Tigers	7	8	16 kg pkg.	62.5	42
19	19	Teatime Chocolate Biscuits	8	3	10 boxes x 12 pieces	9.2	25
20	20	Sir Rodney's Marmalade	8	3	30 gift boxes	81	40
21							

- This is the final display



	ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	1	Chai	10 boxes x 20 bags	39
2	2	Chang	24 - 12 oz bottles	17
3	3	Aniseed Syrup	12 - 550 ml bottles	13
4	4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	5	Chef Anton's Gumbo Mix	36 boxes	0
6	6	Grandma's Boysenberry Spread	12 - 8 oz jars	120
7	7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
8	8	Northwoods Cranberry Sauce	12 - 12 oz jars	6
9	9	Mishi Kobe Niku	18 - 500 g pkgs.	29
10	10	Ikura	12 - 200 ml jars	31
11	11	Queso Cabrales	1 kg pkg.	22
12	12	Queso Manchego La Pastora	10 - 500 g pkgs.	86
13	13	Konbu	2 kg box	24
14	14	Tofu	40 - 100 g pkgs.	35
15	15	Genen Shouyu	24 - 250 ml bottles	39
16	16	Pavlova	32 - 500 g boxes	29
17	17	Alice Mutton	20 - 1 kg tins	0
18	18	Carnarvon Tigers	16 kg pkg.	42
19	19	Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
20	20	Sir Rodney's Marmalade	30 gift boxes	40
21	21	Sir Rodney's Scones	24 pkgs. x 4 pieces	3

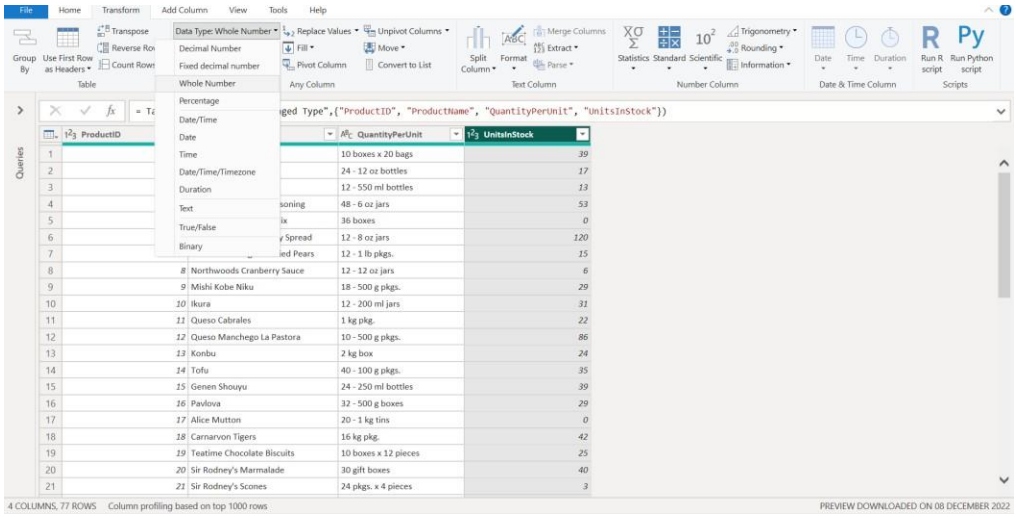
B) Change the data type of the UnitInStock column:

In this step, we will confirm the UnitInStock column's datatype is Whole Number.

When Query Editor connects to data, it reviews each field and to determine the best data type. For the Excel workbook, products in stock will always be a whole number.

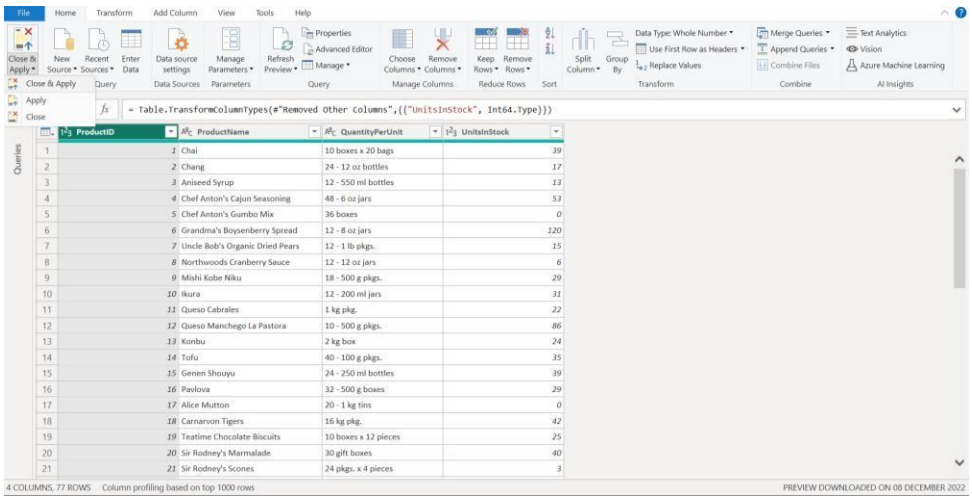
Step 1:

- Select "UnitInStock" column
- Select "Whole Number" from the "Data Type" panel which is in "Transform" ribbon

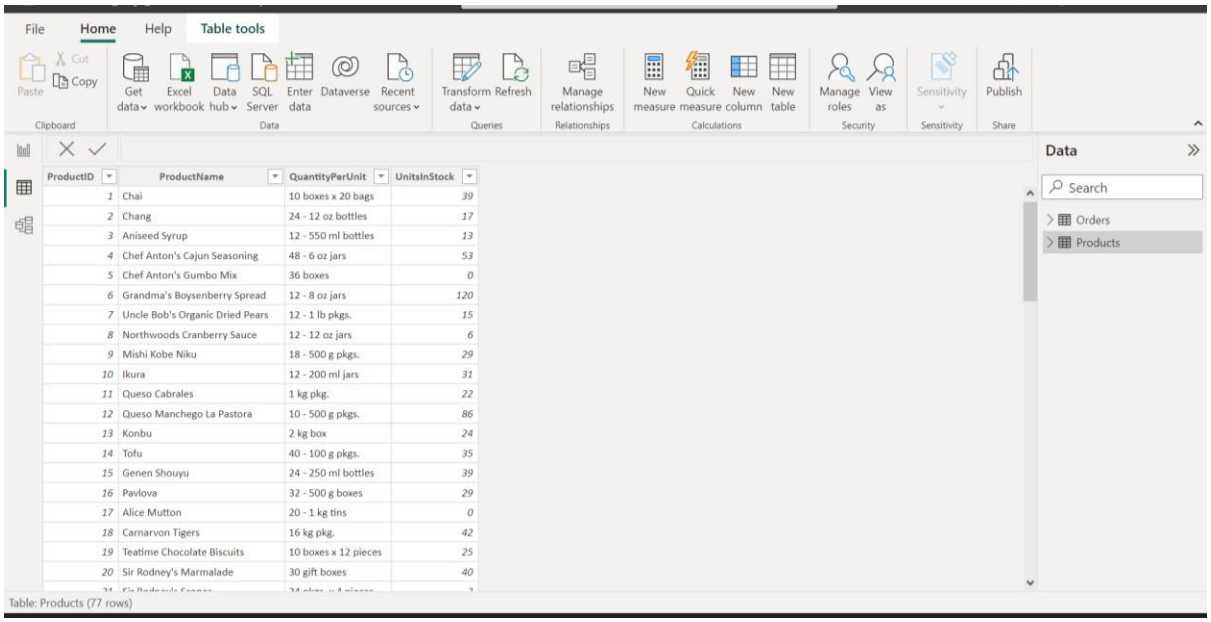


Step 2:

- Applying the necessity changes by selecting "Close & Apply" option from "Home" ribbon.



This is the updated list.



ProductID	ProductName	QuantityPerUnit	UnitsInStock
1	Chai	10 boxes x 20 bags	39
2	Chang	24 - 12 oz bottles	17
3	Aniseed Syrup	12 - 550 ml bottles	13
4	Chef Anton's Cajun Seasoning	48 - 6 oz jars	53
5	Chef Anton's Gumbo Mix	36 boxes	0
6	Grandma's Boysenberry Spread	12 - 8 oz jars	120
7	Uncle Bob's Organic Dried Pears	12 - 1 lb pkgs.	15
8	Northwoods Cranberry Sauce	12 - 12 oz jars	6
9	Mishi Kobe Niku	18 - 500 g pkgs.	29
10	Ikura	12 - 200 ml jars	31
11	Queso Cabrales	1 kg pkg.	22
12	Queso Manchego La Pastora	10 - 500 g pkgs.	86
13	Konbu	2 kg box	24
14	Tofu	40 - 100 g pkgs.	35
15	Genen Shouyu	24 - 250 ml bottles	39
16	Pavlova	32 - 500 g boxes	29
17	Alice Mutton	20 - 1 kg tins	0
18	Carnarvon Tigers	16 kg pkg.	42
19	Teatime Chocolate Biscuits	10 boxes x 12 pieces	25
20	Sir Rodney's Marmalade	30 gift boxes	40

C) Expand the Order_Details table

In this step, you expand the Order_Details table that is related to the Orders table, to combine the ProductID, UnitPrice, and Quantity columns 1from Order_Details into the Orders table.

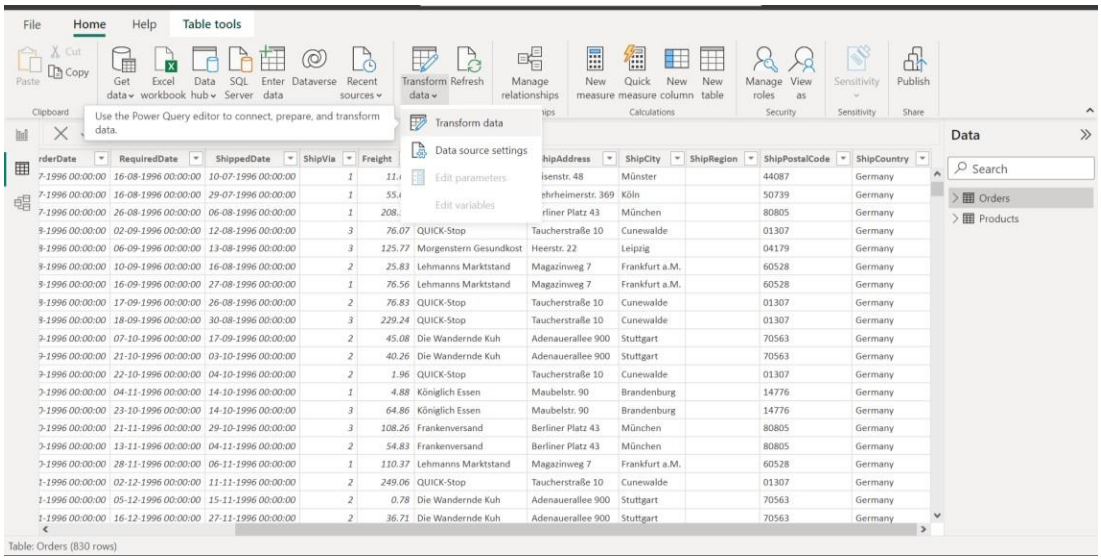
The Expand operation combines columns from a related table into a subject table. When the query runs, rows from the related table (Order_Details) are combined into rows from the subject table (Orders).

After you expand the Order_Details table, three new columns and additional rows are added to the Orders table, one for each row in the nested or related table.

The Orders table contains a reference to a Details table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query.

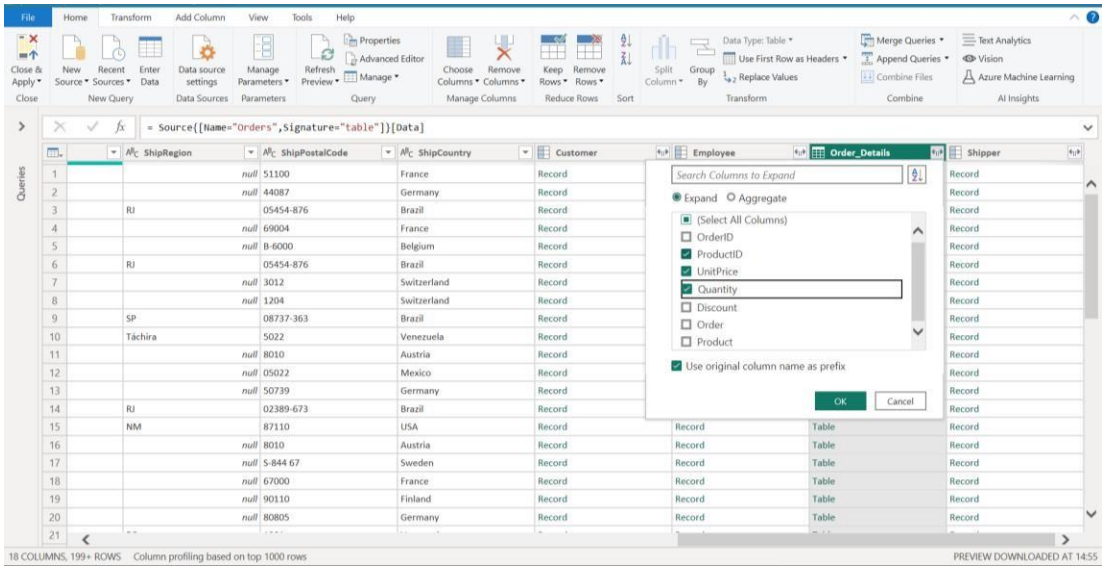
Step 1:

- From the "Home" ribbon, select "Transform data" option

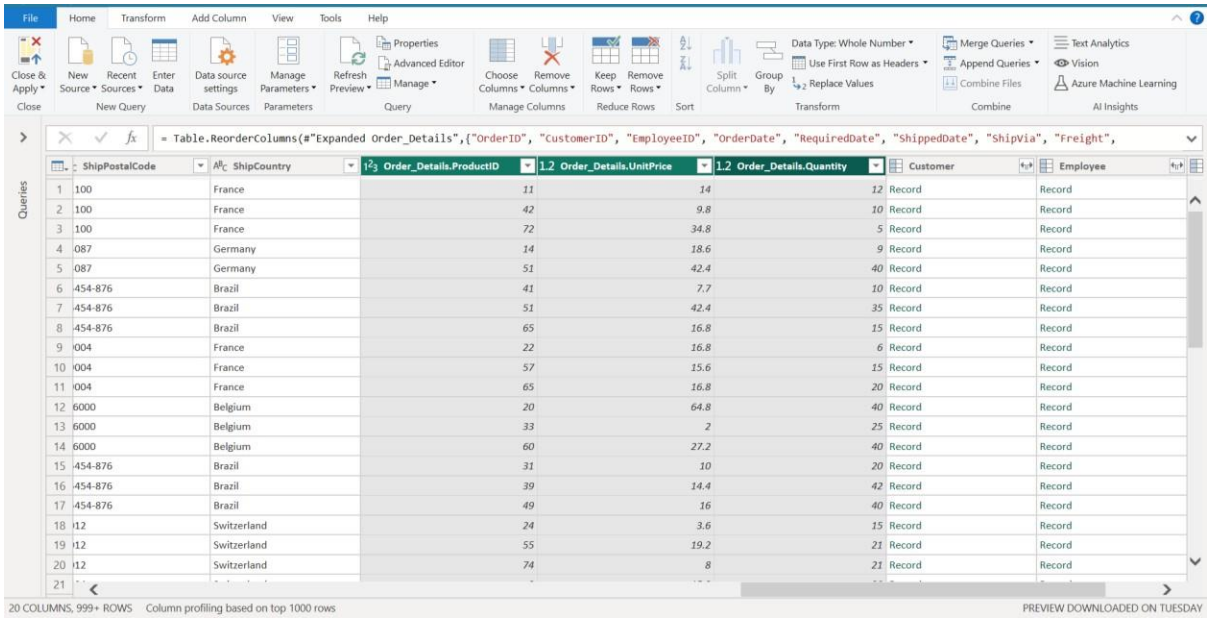


Step 2:

- From the columns displayed → go to Order_Details → click on the "expand" icon beside it → select the column of interest (ProductID, UnitPrice, Quantity) → click on "OK"



- All the columns will be displayed



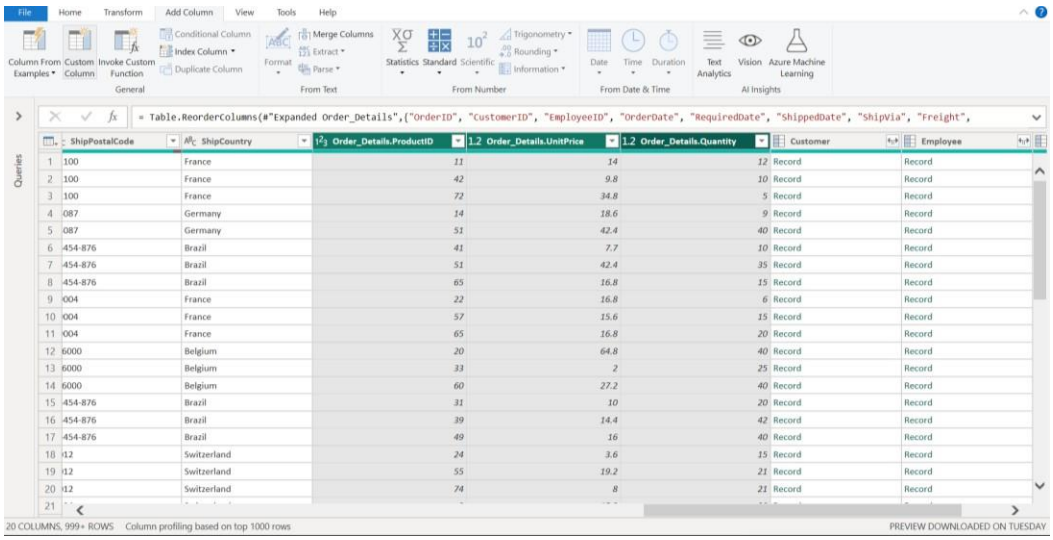
D) Calculate the line total for each Order_Details row

In this step, you create a Custom Column to calculate the line total for each Order_Details row.

One can to create calculations based on the columns you are importing, so you can enrich the data that you connect to.

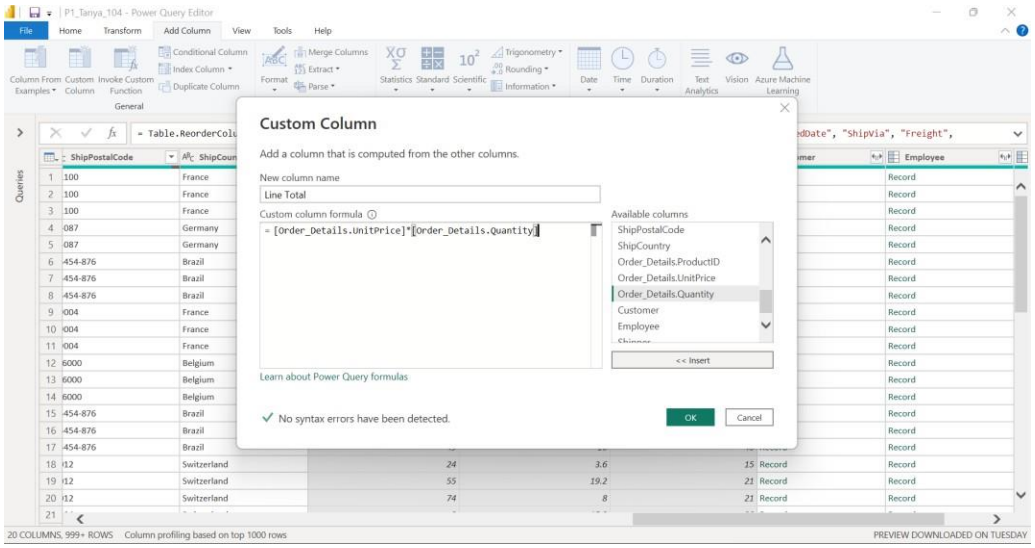
Step 1:

- From the "Add column" ribbon, select "Custom Column".

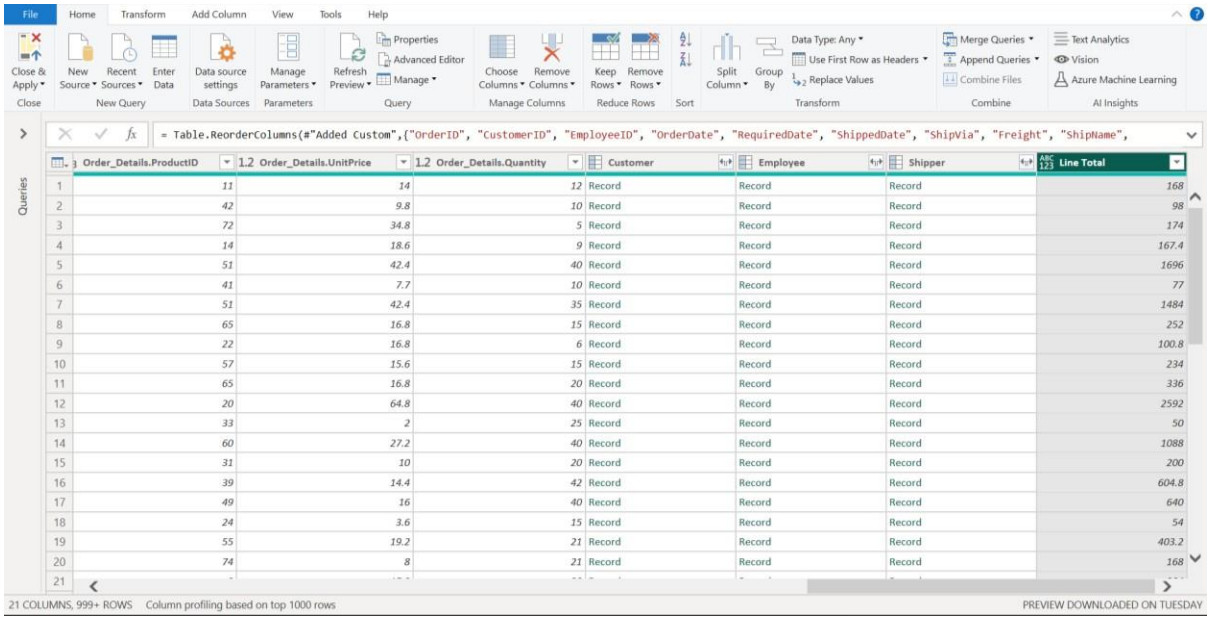


Step 2:

- Change the New column name as "Line Total".
- For formula, select "Order_Details.UnitPrice" from available columns list and click on "insert". Insert an "asterisk (*)". Select "Order_Details.Quantity" from available columns list and click on "insert".
- Select "OK"



- "Line total" column will be displayed.



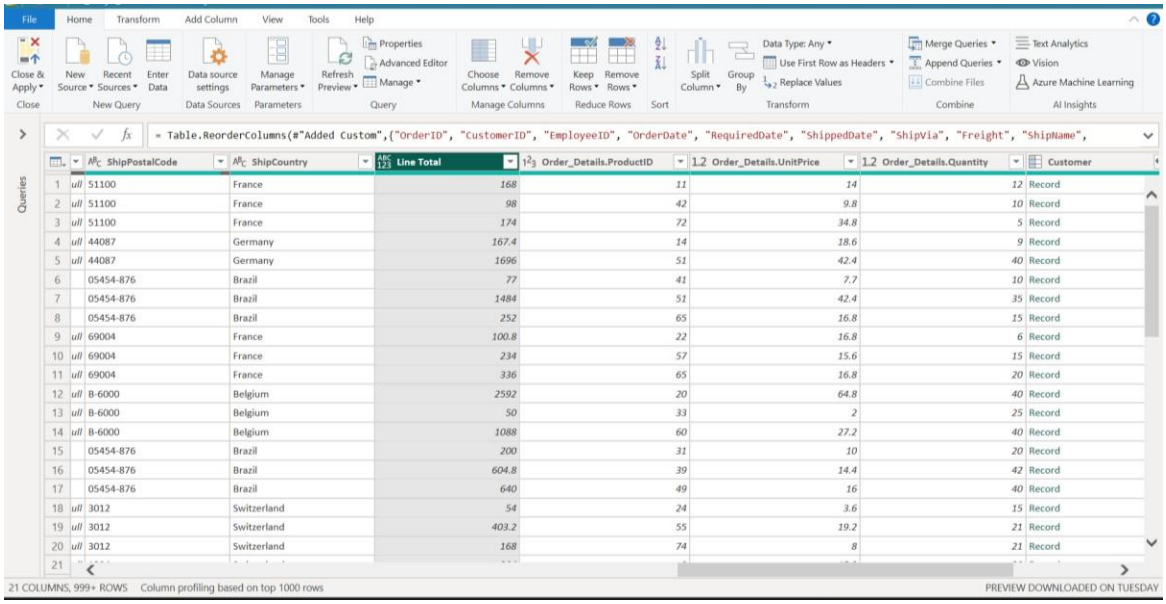
	Order_Details.ProductID	Order_Details.UnitPrice	Order_Details.Quantity	Customer	Employee	Shipper	Line Total
1	11	14	12	Record	Record	Record	168
2	42	9.8	10	Record	Record	Record	98
3	72	34.8	5	Record	Record	Record	174
4	14	18.6	9	Record	Record	Record	167.4
5	51	42.4	40	Record	Record	Record	1696
6	41	7.7	10	Record	Record	Record	77
7	51	42.4	35	Record	Record	Record	1484
8	65	16.8	15	Record	Record	Record	252
9	22	16.8	6	Record	Record	Record	100.8
10	57	15.6	15	Record	Record	Record	234
11	65	16.8	20	Record	Record	Record	336
12	20	64.8	40	Record	Record	Record	2592
13	33	2	25	Record	Record	Record	50
14	60	27.2	40	Record	Record	Record	1088
15	31	10	20	Record	Record	Record	200
16	39	14.4	42	Record	Record	Record	604.8
17	49	16	40	Record	Record	Record	640
18	24	3.6	15	Record	Record	Record	54
19	55	19.2	21	Record	Record	Record	403.2
20	74	8	21	Record	Record	Record	168
21							

E) Rename and reorder columns in the query

In this step, we will rename the necessary columns and change their orders for making a better model.

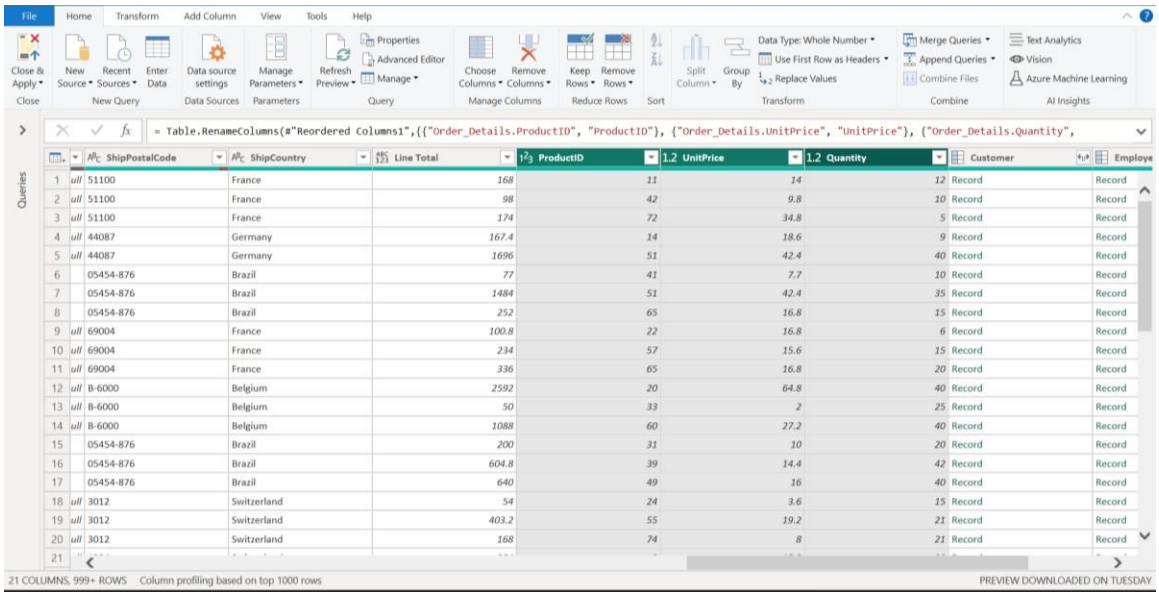
Step 1:

- Drag “Line Total” column towards the left side to reorder the data.

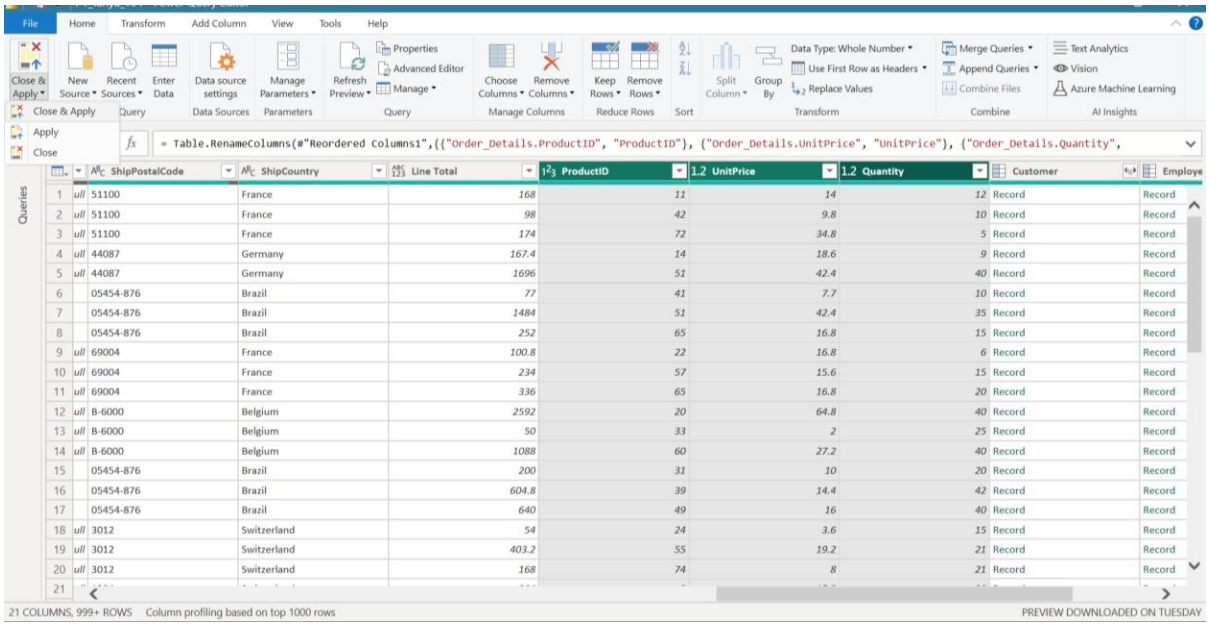


Step 2:

- Rename Order_Details columns by removing the prefix – Order_Details from their heading i.e., changing Order_Details.ProductID to ProductID, Order_Details.UnitPrice to UnitPrice and Order_Details.Quantity to Quantity.



- Applying the necessity changes by selecting “Close & Apply” option from “Home” ribbon.



	ShipPostalCode	ShipCountry	Line Total	ProductID	UnitPrice	Quantity	Customer	Employe
1	null	51100	France	168	11	14	12	Record	Record
2	null	51100	France	98	42	9.8	10	Record	Record
3	null	51100	France	174	72	34.8	5	Record	Record
4	null	44087	Germany	167.4	14	18.6	9	Record	Record
5	null	44087	Germany	1696	51	42.4	40	Record	Record
6		05454-876	Brazil	77	41	7.7	10	Record	Record
7		05454-876	Brazil	1484	51	42.4	35	Record	Record
8		05454-876	Brazil	252	65	16.8	15	Record	Record
9	null	69004	France	100.8	22	16.8	6	Record	Record
10	null	69004	France	234	57	15.6	15	Record	Record
11	null	69004	France	336	65	16.8	20	Record	Record
12	null	B-6000	Belgium	2592	20	64.8	40	Record	Record
13	null	B-6000	Belgium	50	33	2	25	Record	Record
14	null	B-6000	Belgium	1088	60	27.2	40	Record	Record
15		05454-876	Brazil	200	31	10	20	Record	Record
16		05454-876	Brazil	604.8	39	14.4	42	Record	Record
17		05454-876	Brazil	640	49	16	40	Record	Record
18	null	3012	Switzerland	54	24	3.6	15	Record	Record
19	null	3012	Switzerland	403.2	55	19.2	21	Record	Record
20	null	3012	Switzerland	168	74	8	21	Record	Record
21									

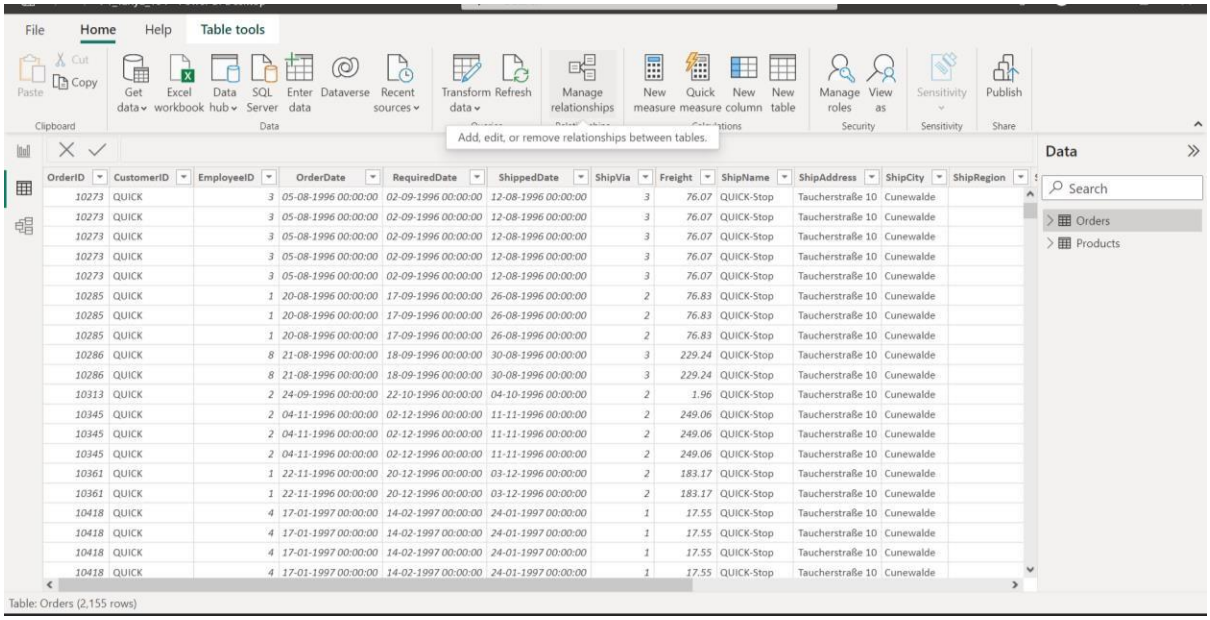
F) Combine the Products and Total Sales queries

In this step, you confirm that a relationship is established between the Products and Total Sales

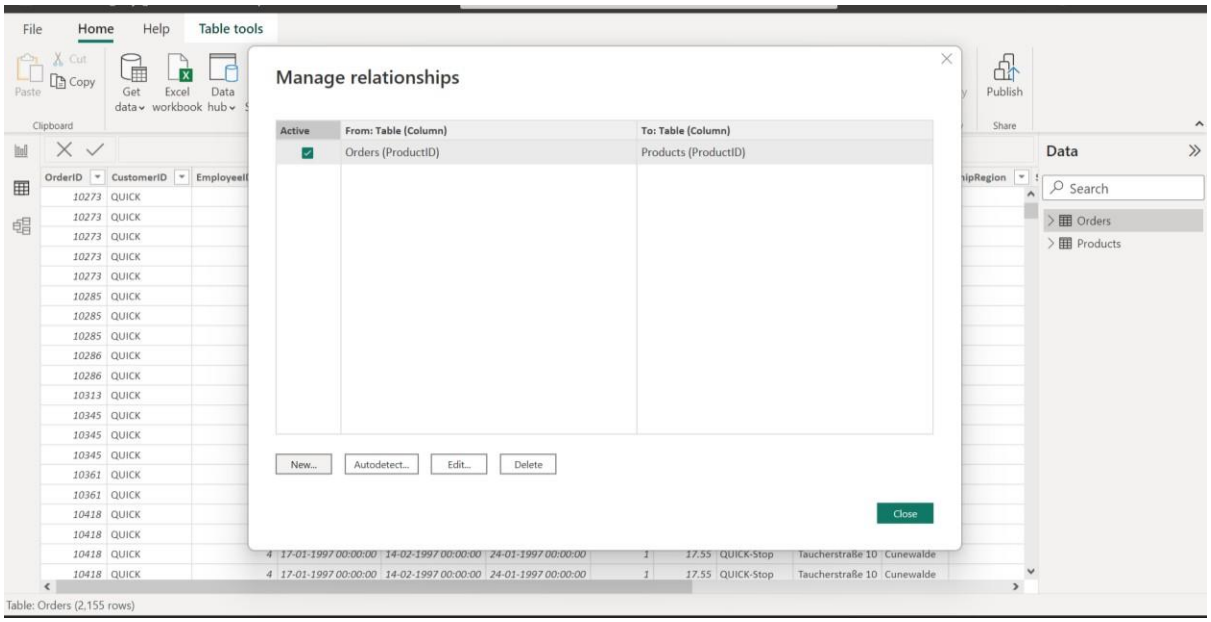
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets we have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

Step 1:

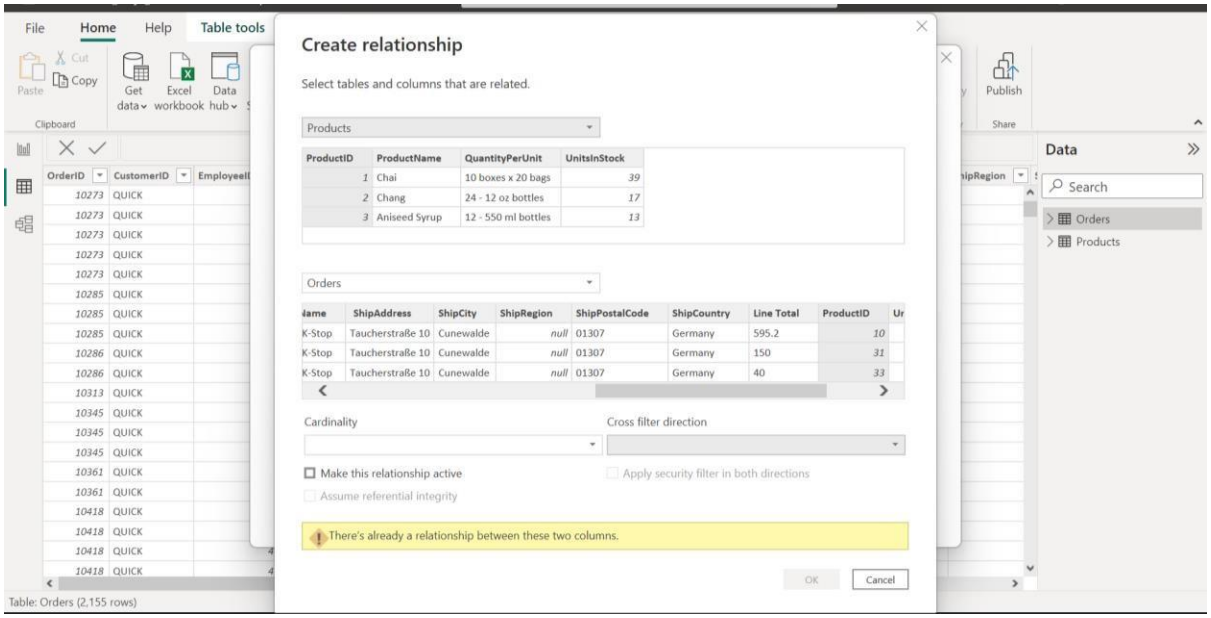
- Select “Manage Relationships” from the “Home” ribbon.



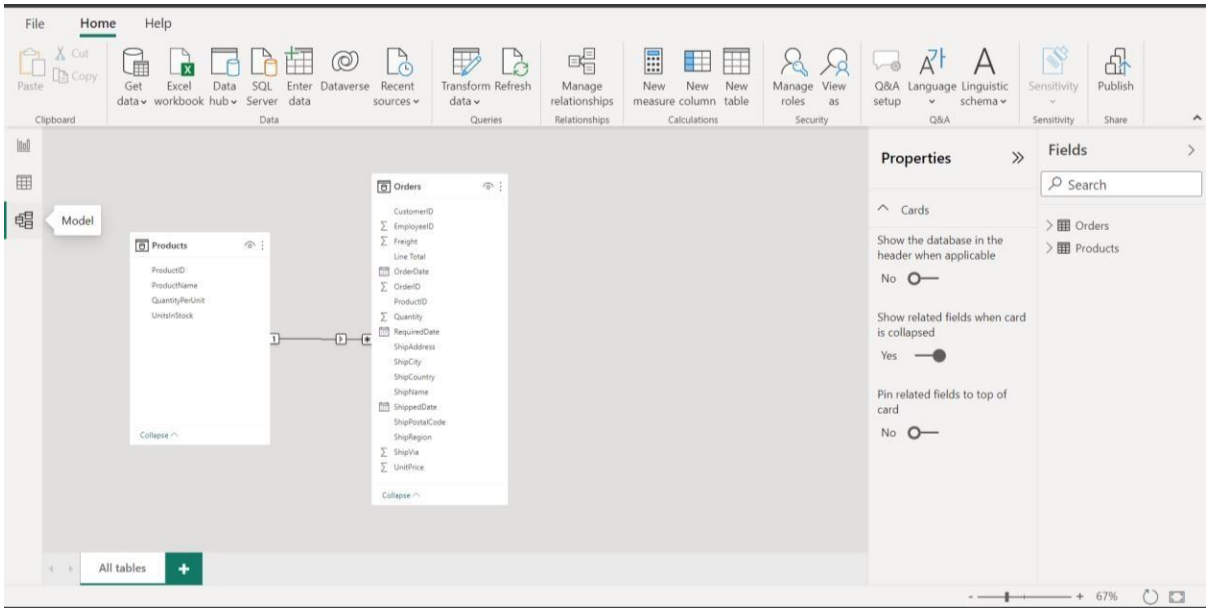
- As displayed a relationship is created by default between “Orders and Products”.
- To create a relation between two data, click on “New”



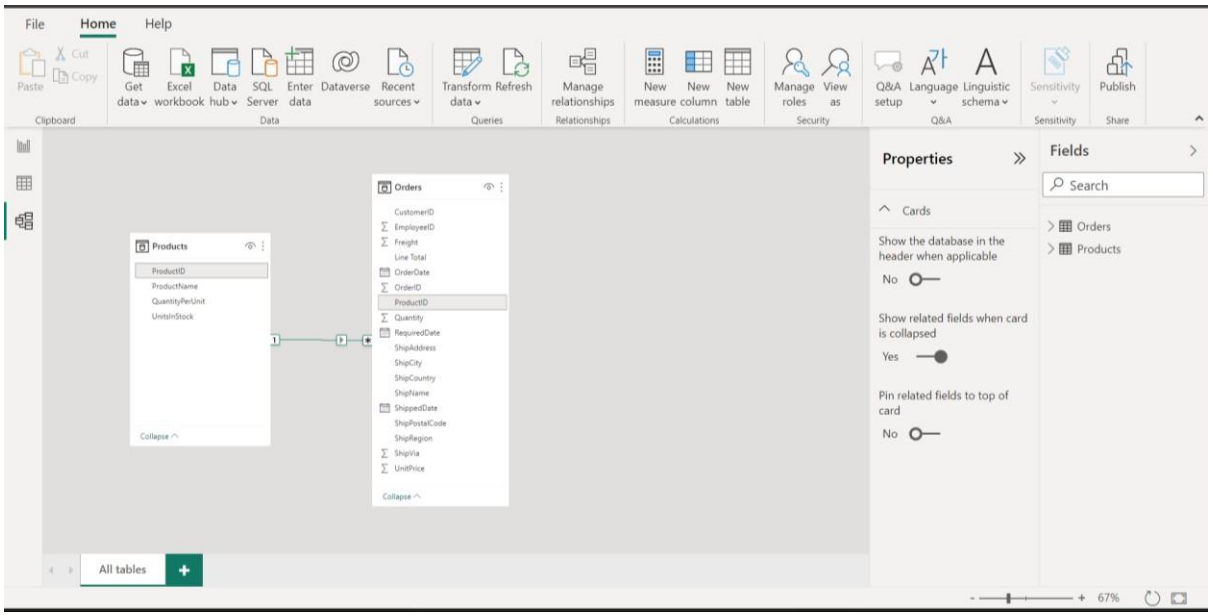
- Select “Products” in from table and “Orders” in to table.
- Here, the relationship is already created by default so we cannot create the same relationship.



- From the left panel, select “Model”
- Here, you can view the relationship created.



- When you click on relationship, you can view that “ProductID” is in common to create this relationship model.



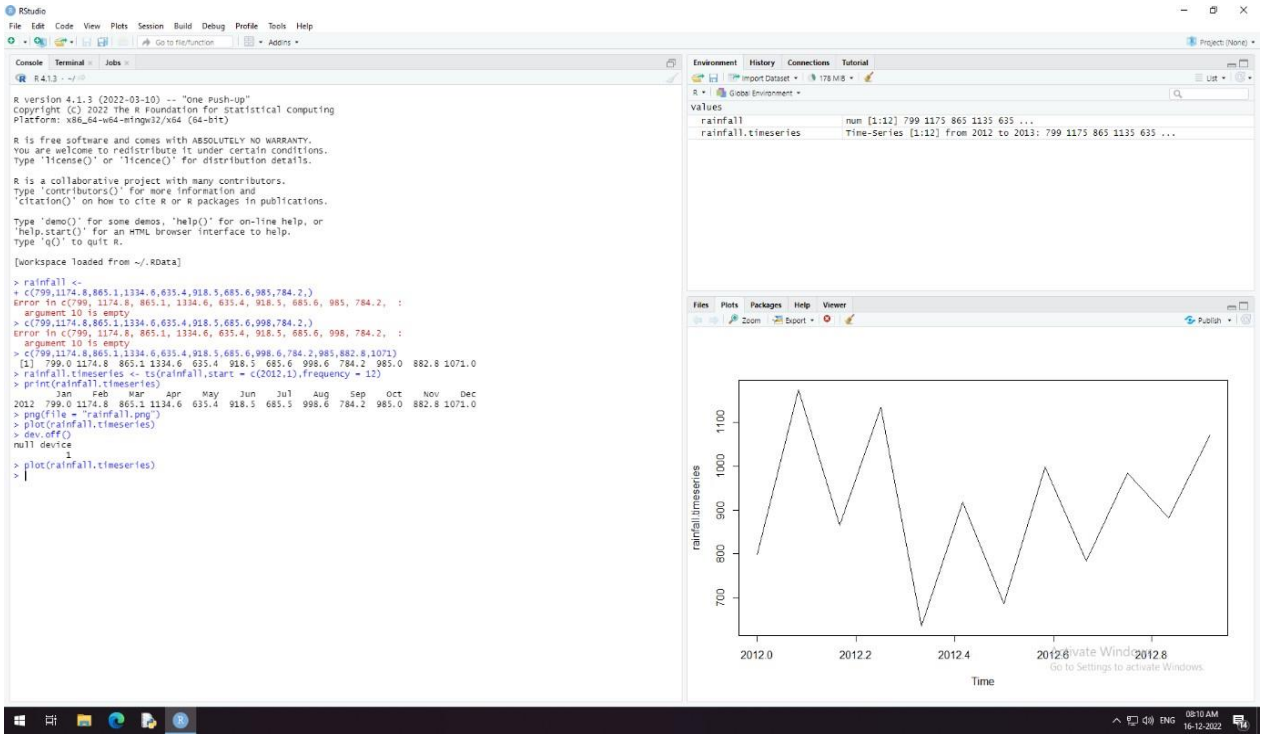
Practical 3

Implementation of Classification Algorithm in R Programming.

- Consider the annual rainfall details at a place starting from January 2012. We create an R time series object for a period of 12 months and plot it.

```
# Get the data points in form of a R vector.
rainfall <-c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)
# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
# Print the timeseries data.
print(rainfall.timeseries)
# Give the chart file a name.
png(file = "rainfall.png")
# Plot a graph of the time series.
plot(rainfall.timeseries)
# Save the file.
dev.off()
```

Output:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Jobs
R 4.1.3 · ~/
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
> rainfall <-
+ c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,985,784.2,)
Error in c(799, 1174.8, 865.1, 1334.6, 635.4, 918.5, 685.6, 985, 784.2, :
argument 10 is empty
> c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,998,784.2,)
Error in c(799, 1174.8, 865.1, 1334.6, 635.4, 918.5, 685.6, 998, 784.2, :
argument 10 is empty
> c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,998.6,784.2,985,882.8,1071)
[1] 799.0 1174.8 865.1 1334.6 635.4 918.5 685.6 998.6 784.2 985.0 882.8 1071.0
> rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
> print(rainfall.timeseries)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2012 799.0 1174.8 865.1 1334.6 635.4 918.5 685.5 998.6 784.2 985.0 882.8 1071.0
> png(file = "rainfall.png")
> plot(rainfall.timeseries)
> dev.off()
null device
1
> plot(rainfall.timeseries)
Environment History Connections Tutorial
Import Dataset
Global Environment
Values
rainfall num [1:12] 799 1175 865 1135 635 ...
rainfall.timeseries Time-Series [1:12] from 2012 to 2013: 799 1175 865 1135 635 ...
Files Plots Packages Help Viewer
Zoom Export Publish
rainfall.timeseries
700 800 900 1000 1100
2012.0 2012.2 2012.4 2012.6 2012.8
Time
Activate Windows
Go to Settings to activate Windows.
ENG 08:10 AM 16-12-2022

Practical 4

K-means Clustering using R Programming.

K-means clustering (MacQueen 1967) is one of the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups (i.e. k clusters), where k represents the number of groups pre-specified by the analyst. It classifies objects in multiple groups (i.e., clusters), such that objects within the same cluster are as similar as possible (i.e., high intra-class similarity), whereas objects from different clusters are as dissimilar as possible (i.e., low inter-class similarity). In k-means clustering, each cluster is represented by its center (i.e, centroid) which corresponds to the mean of points assigned to the cluster.The algorithm tries to find groups by minimizing the distance between the observations, called local optimal solutions. The distances are measured based on the coordinates of the observations. For instance, in a two-dimensional space, the coordinates are simple

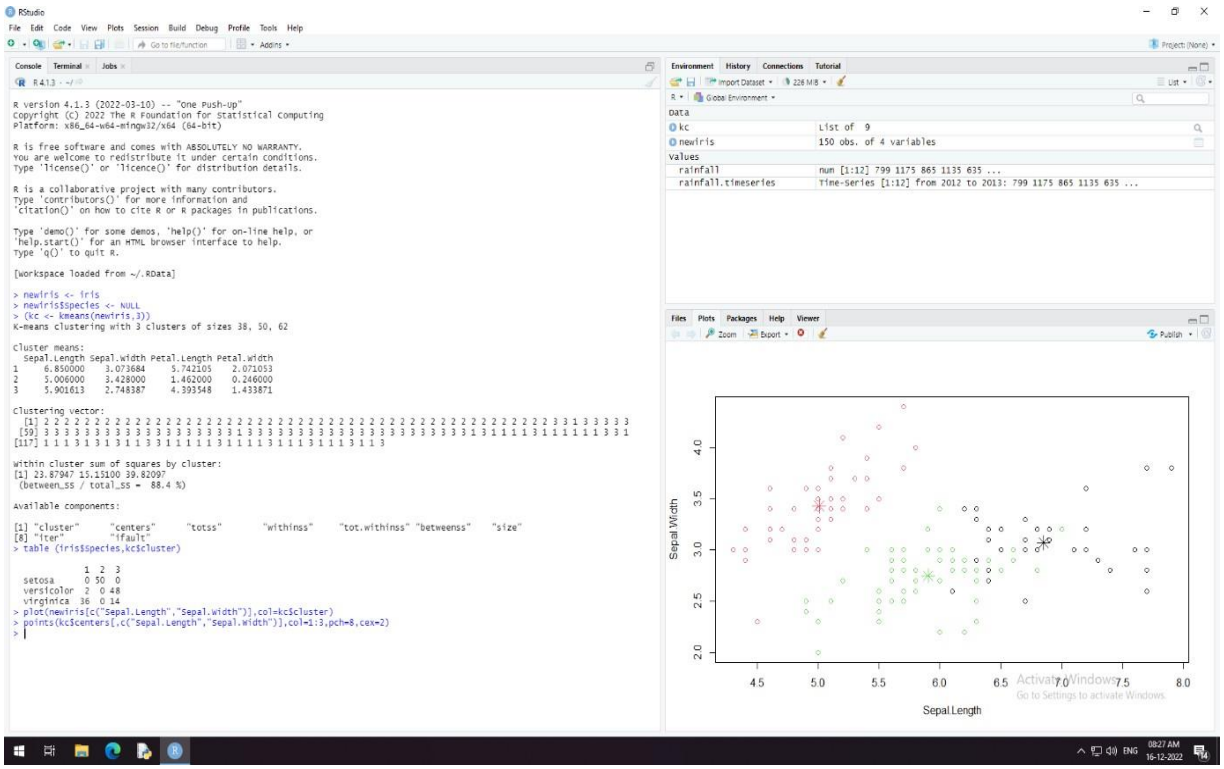
Theory

- K-Means clustering groups the data on similar groups. The algorithm is as follows:
- Choose the number K clusters.
- Select at random K points, the centroids(Not necessarily from the given data).
- Assign each data point to closest centroid that forms K clusters.
- Compute and place the new centroid of each centroid.
- Reassign each data point to new cluster.

The algorithm works as follow:

- Step 1: Choose groups in the feature plan randomly
- Step 2: Minimize the distance between the cluster center and the different observations (centroid). It results in groups with observations
- Step 3: Shift the initial centroid to the mean of the coordinates within a group.
- Step 4: Minimize the distance according to the new centroids. New boundaries are created. Thus, observations will move from one group to another.
- Repeat until no observation changes groups

Output:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Jobs
R version 4.1.3 (2022-03-10) -- "One Push-Up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris,3))
K-means clustering with 3 clusters of sizes 38, 50, 62
Cluster means:
Sepal.Length Sepal.Width Petal.Length Petal.Width
1 6.850000 3.073684 5.742105 2.071053
2 5.006000 3.428000 1.462000 0.246000
3 5.901613 2.748387 4.393548 1.433871
Clustering vector:
Within cluster sum of squares by cluster:
[1] 23.87947 15.15100 39.82097
(between_SS / total_SS = 88.4 %)
Available components:
[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size"
[8] "iter" "ifault"
> table (iris$Species,kc$cluster)
1 2 3
setosa 0 50 0
versicolor 2 0 48
virginica 36 0 14
> plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)
> points(kc$centers[,c("Sepal.Length","Sepal.width")],col=1:3,pch=8,cex=2)
Environment History Connections Tutorial
Import Dataset
Global Environment
Data
kc List of 9
newiris 150 obs. of 4 variables
Values
rainfall num [1:12] 799 1175 865 1135 635 ...
rainfall.timeseries Time-Series [1:12] from 2012 to 2013: 799 1175 865 1135 635 ...
Files Plots Packages Help Viewer
Zoom Export
Publish
Sepal.Width
Sepal.Length
2.0 2.5 3.0 3.5 4.0
4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
Activate Windows
Go to Settings to activate Windows.
ENG 08:27 AM 16-12-2022

Practical 5

Prediction Using Linear Regression in R Programming.

In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

y = ax + b is an equation for linear regression.

Where, y is the response variable, x is the predictor variable and a and b are constants which are called the coefficients.

A simple example of regression is predicting weight of a person when his height is known. To do this we need to have the relationship between height and weight of a person.

The steps to create the relationship is −

- Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
- Create a relationship model using the lm() functions in R.
- Find the coefficients from the model created and create the mathematical equation using these
- Get a summary of the relationship model to know the average error in prediction. Also called residuals.
- To predict the weight of new persons, use the predict() function in R.

Input Data

Below is the sample data representing the observations −

Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

Values of weight.

63, 81, 56, 91, 47, 57, 76, 72, 62, 48

lm() Function

This function creates the relationship model between the predictor and the response variable.

Syntax

The basic syntax for lm() function in linear regression is −

lm(formula,data)

Following is the description of the parameters used −

- formula is a symbol presenting the relation between x and y.
- data is the vector on which the formula will be applied.

Create Relationship Model & get the Coefficients

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

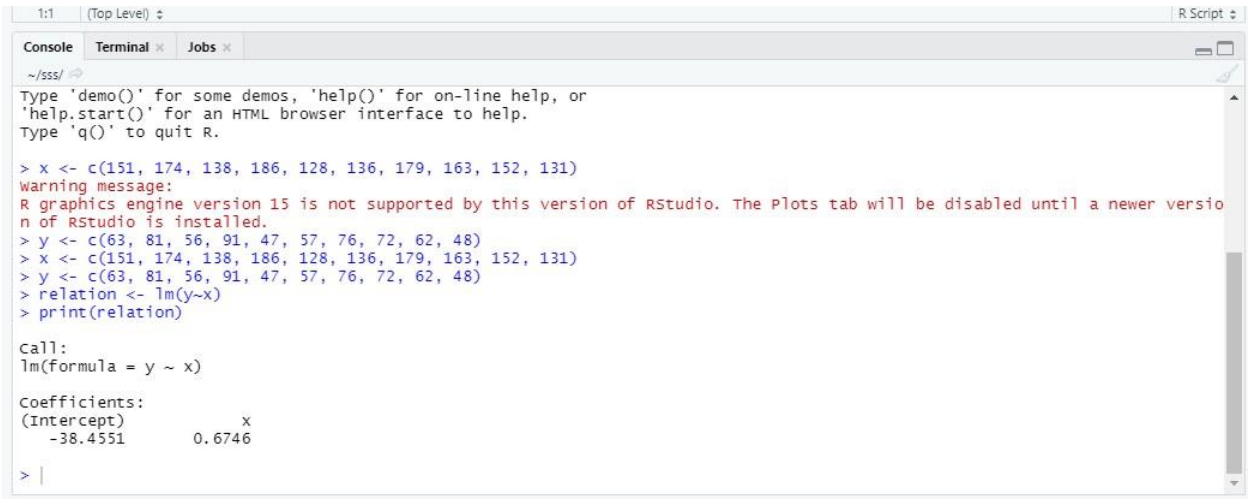
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

Apply the lm() function.

relation <- lm(y~x)

print(relation)

When we execute the above code, it produces the following result –



Get the Summary of the Relationship

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

Apply the lm() function.

relation <- lm(y~x)

print(summary(relation))

When we execute the above code, it produces the following result –

```
> y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> relation <- lm(y~x)
> print(summary(relation))

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06

> |
```

predict() Function

Syntax

The basic syntax for predict() in linear regression is −

predict(object, newdata)

Following is the description of the parameters used −

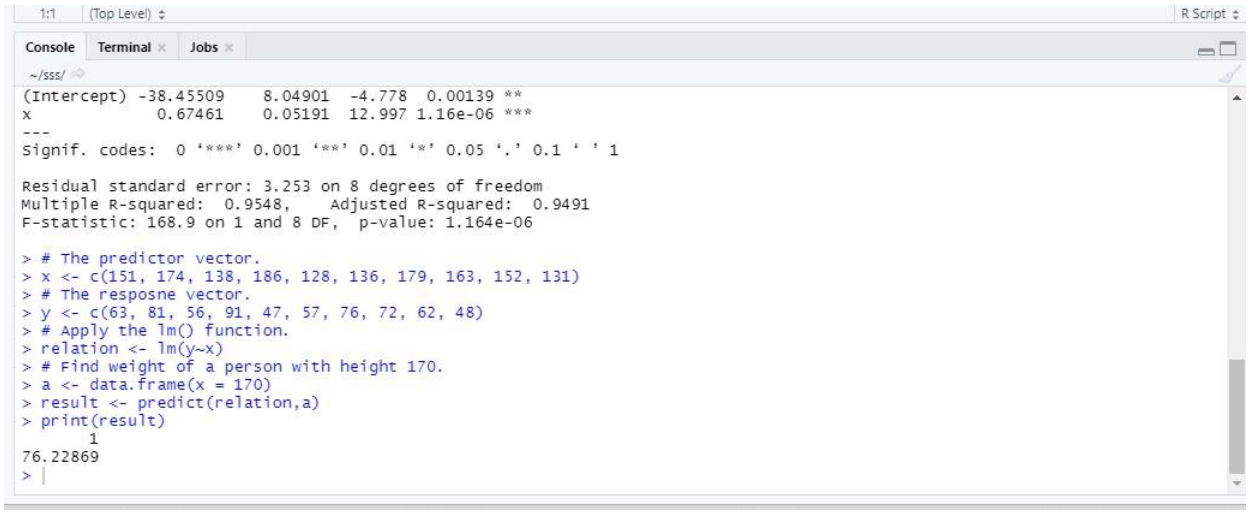
- object is the formula which is already created using the lm() function.
- newdata is the vector containing the new value for predictor variable.

Predict the weight of new persons

```
# The predictor vector.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

# Find weight of a person with height 170.
a <- data.frame(x = 170)
result <- predict(relation,a)
print(result)
```

Result:

```
    1
76.22869
```



Visualize the Regression Graphically

Create the predictor and response variable.

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

relation <- lm(y~x)

Give the chart file a name.

png(file = "linearregression.png")

Plot the chart.

plot(y,x,col = "blue",main = "Height & Weight Regression",

abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")

Save the file.

dev.off()

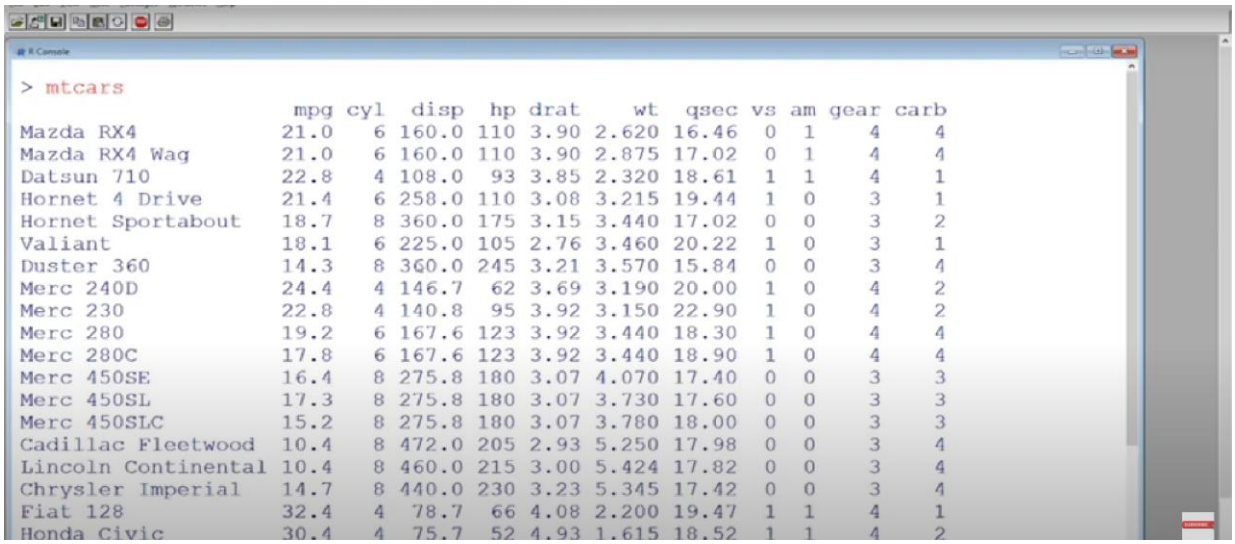
Output:


sss - RStudio
# Create the predictor and response variable.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)
# Give the chart file a name.
png(file = "linearregression.png")
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")
# Save the file.
dev.off()
> result <- predict(relation,a)
> print(result)
76.32869
> dev.off()
null device
R Graphics: Device 2 (ACTIVE)
File History Resize
Height & Weight Regression
Weight in Kg
cm
130
140
150
160
170
180
50
60
70
80
90

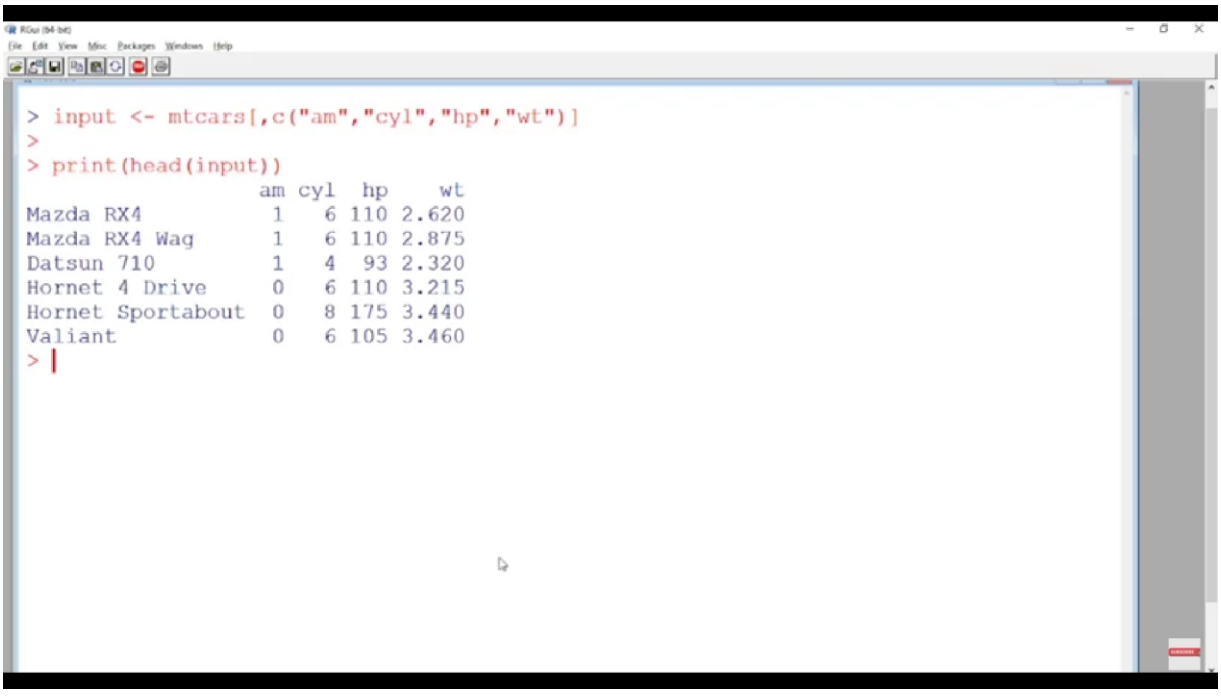
Practical 6

Perform the logistic regression on the given data warehouse data..

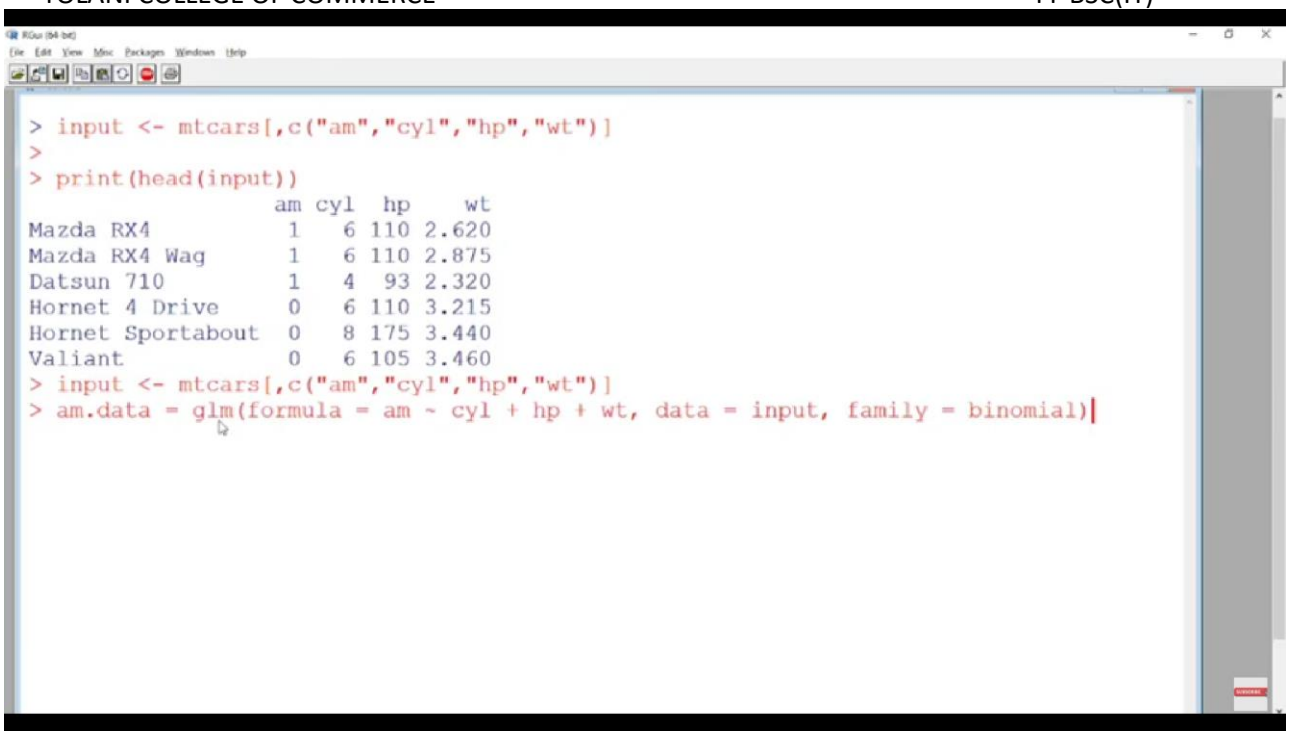
Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.



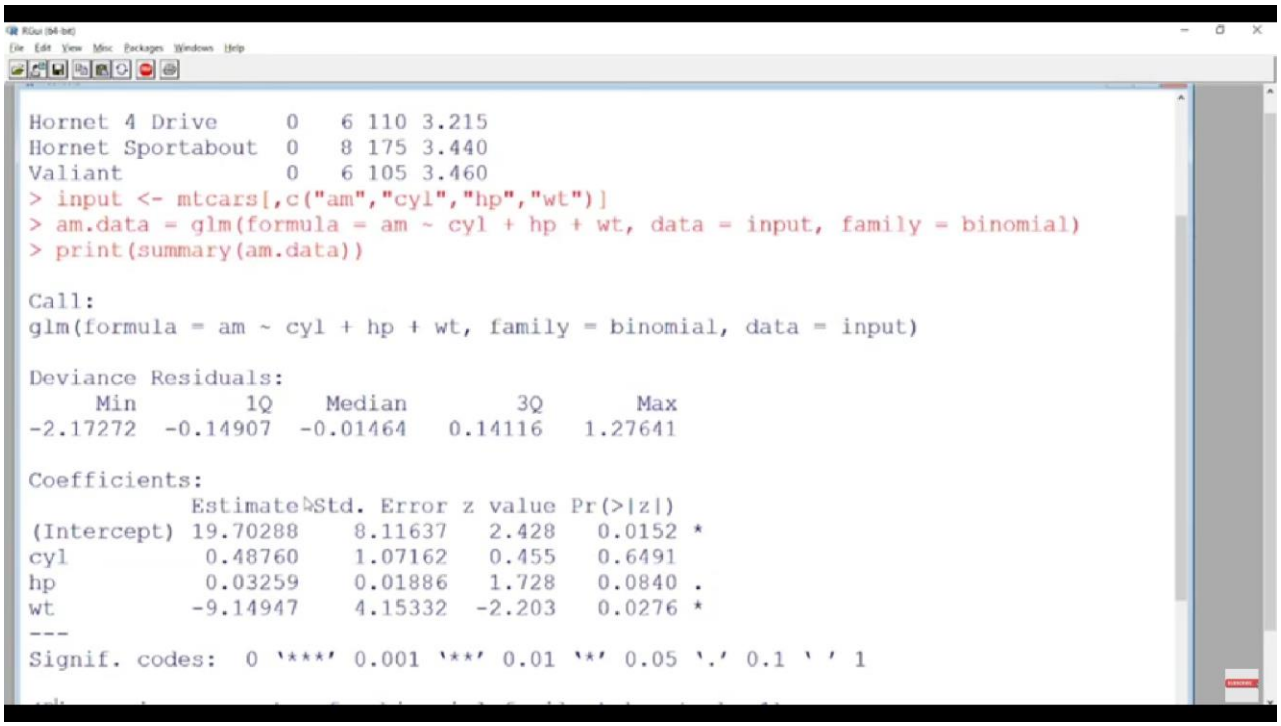
```
> mtcars
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```



```
> input <- mtcars[,c("am","cyl","hp","wt")]
>
> print(head(input))
                  am cyl  hp    wt
Mazda RX4          1   6 110 2.620
Mazda RX4 Wag      1   6 110 2.875
Datsun 710         1   4  93 2.320
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
>
```

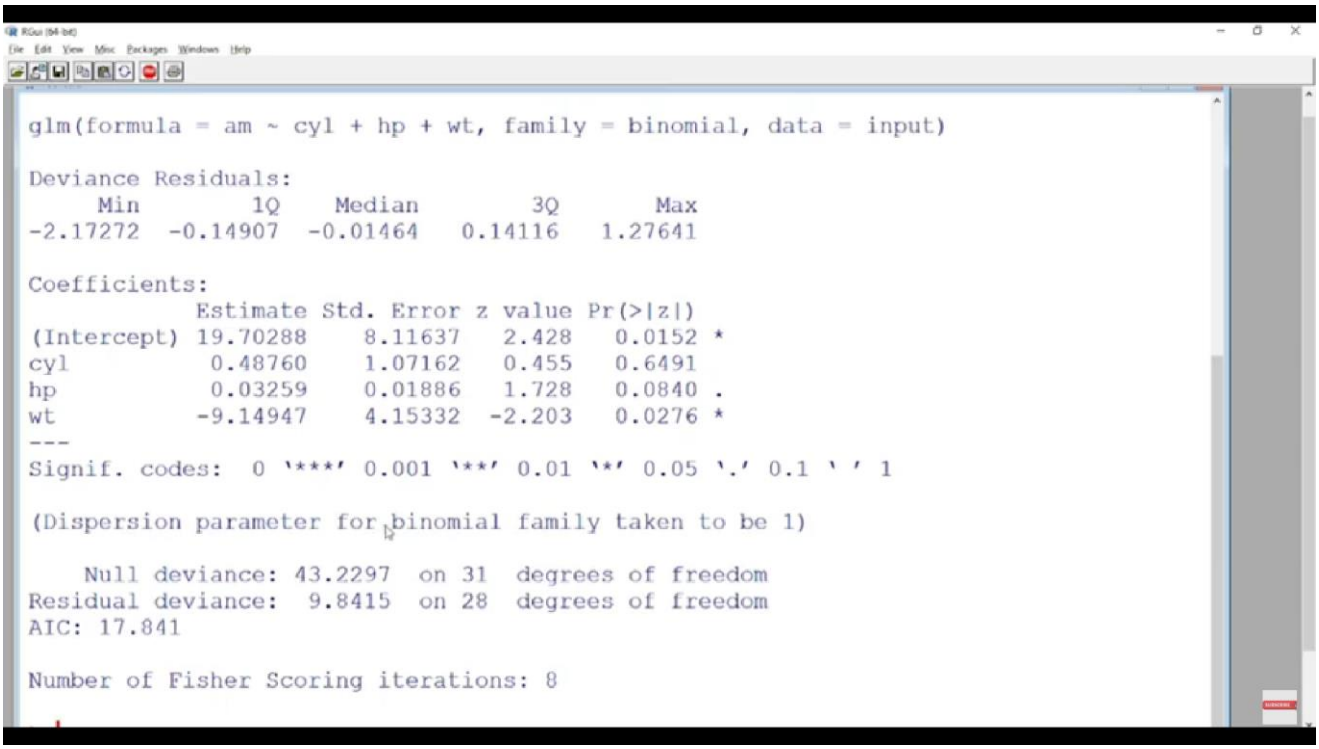


```
> input <- mtcars[,c("am","cyl","hp","wt")]
>
> print(head(input))
                  am cyl  hp    wt
Mazda RX4          1   6 110 2.620
Mazda RX4 Wag      1   6 110 2.875
Datsun 710         1   4  93 2.320
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
> input <- mtcars[,c("am","cyl","hp","wt")]
> am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)
```



```
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
> input <- mtcars[,c("am","cyl","hp","wt")]
> am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)
> print(summary(am.data))

Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17272  -0.14907  -0.01464   0.14116   1.27641

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Output

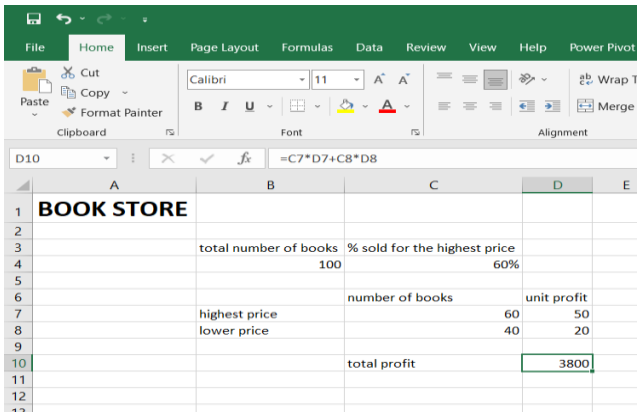


```
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17272  -0.14907  -0.01464   0.14116   1.27641

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.2297  on 31  degrees of freedom
Residual deviance:  9.8415  on 28  degrees of freedom
AIC: 17.841

Number of Fisher Scoring iterations: 8
```

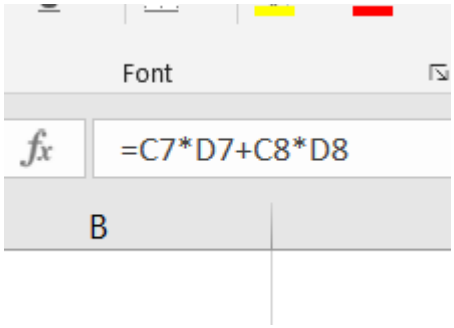
Practical 7

Apply What – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data.

A book store and have 100 books in storage. You sell a certain % for the highest price of $50 and a certain % for the lower price of $20.



If you sell 60% for the highest price, cell D10 calculates a total profit of 60 * $50 + 40 * $20 = $3800.



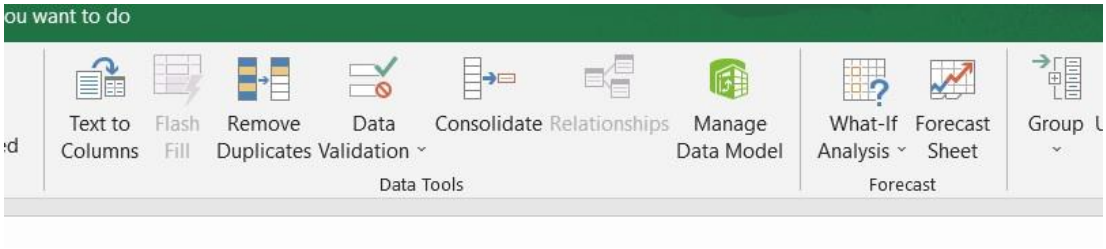
Create Different Scenarios But what if you sell 70% for the highest price? And what if you sell 80% for the highest price? Or 90%, or even 100%? Each different percentage is a different scenario. You can use the Scenario Manager to create these scenarios. Note: You can simply type in a different percentage into cell C4 to see the corresponding result of a scenario in cell D10. However, what-if analysis enables you to easily compare the results of different scenarios. Read on.

For C8 this formula is applied 100-60(B4-C7)

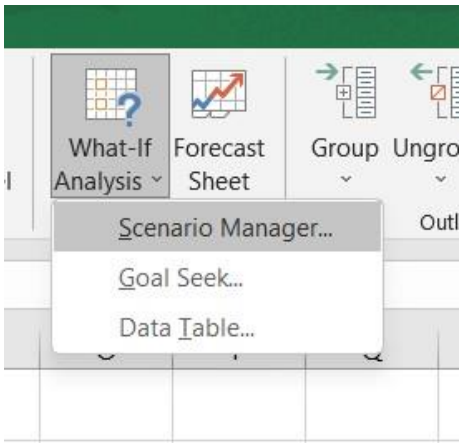
C8 =B4-C7

	A	B	C	D	E	F
1	book store					
2						
3		total no of books	% for the highest			
4		100	60%			
5						
6			number of books	unit profit		
7		highest price	60	50		
8		lower price	40	20		
9				total	3800	
10						
11						
12						

1. On the Data tab, in the Forecast group, click What-If Analysis

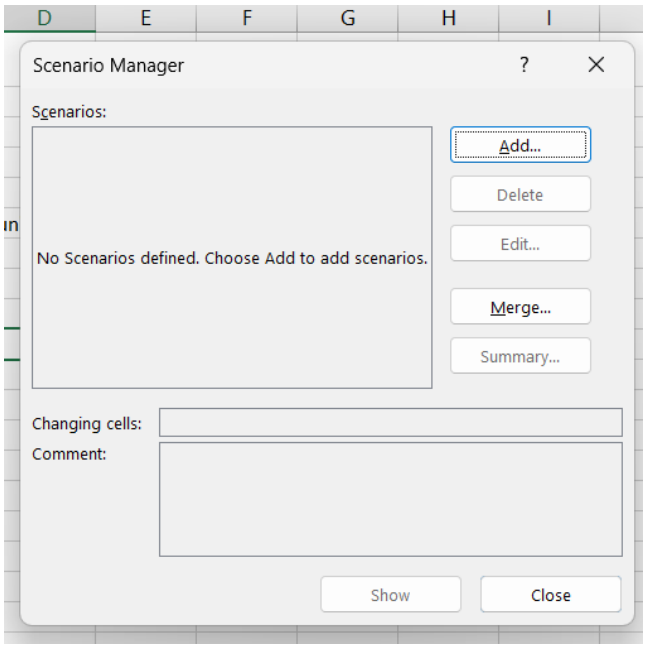


2. Click Scenario Manager.

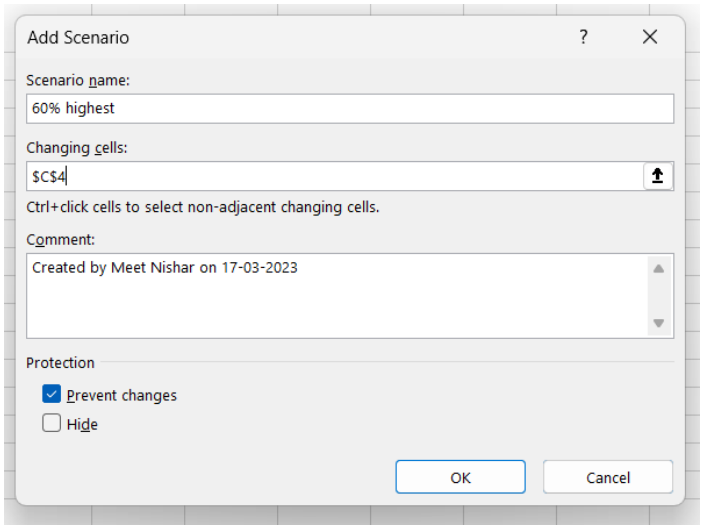


The Scenario Manager dialog box appears.

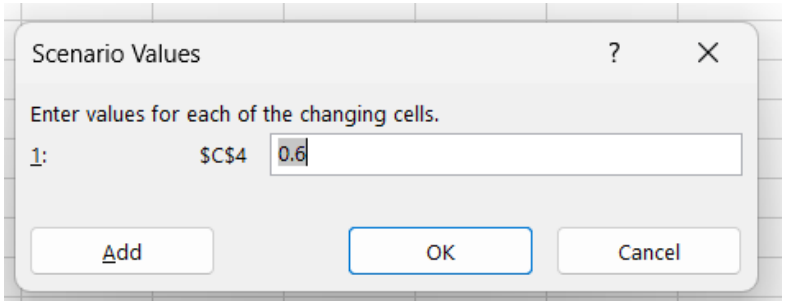
3. Add a scenario by clicking on Add.



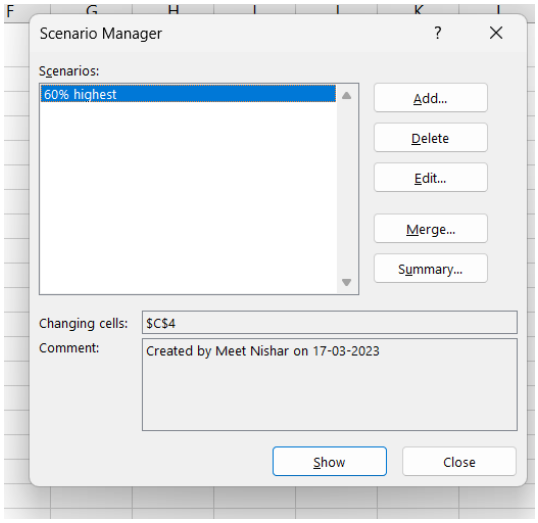
4. Type a name (60% highest), select cell C4 (% sold for the highest price) for the Changing cells and Click on OK.



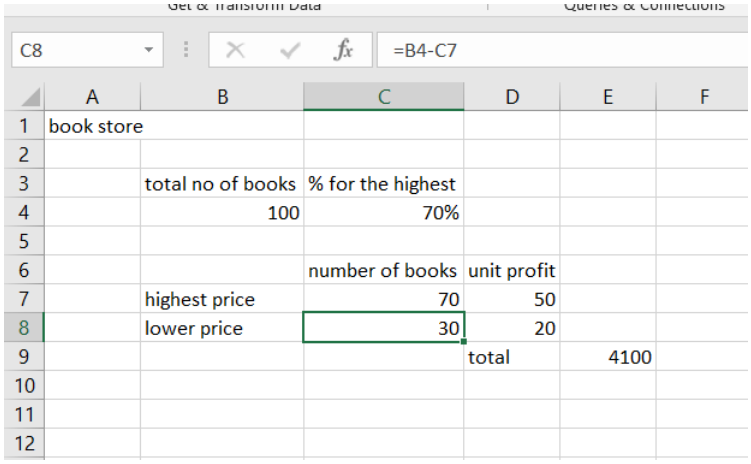
5. Enter the corresponding value 0.6 and click on OK again.



6. 60% highest will be added to the Dialog box



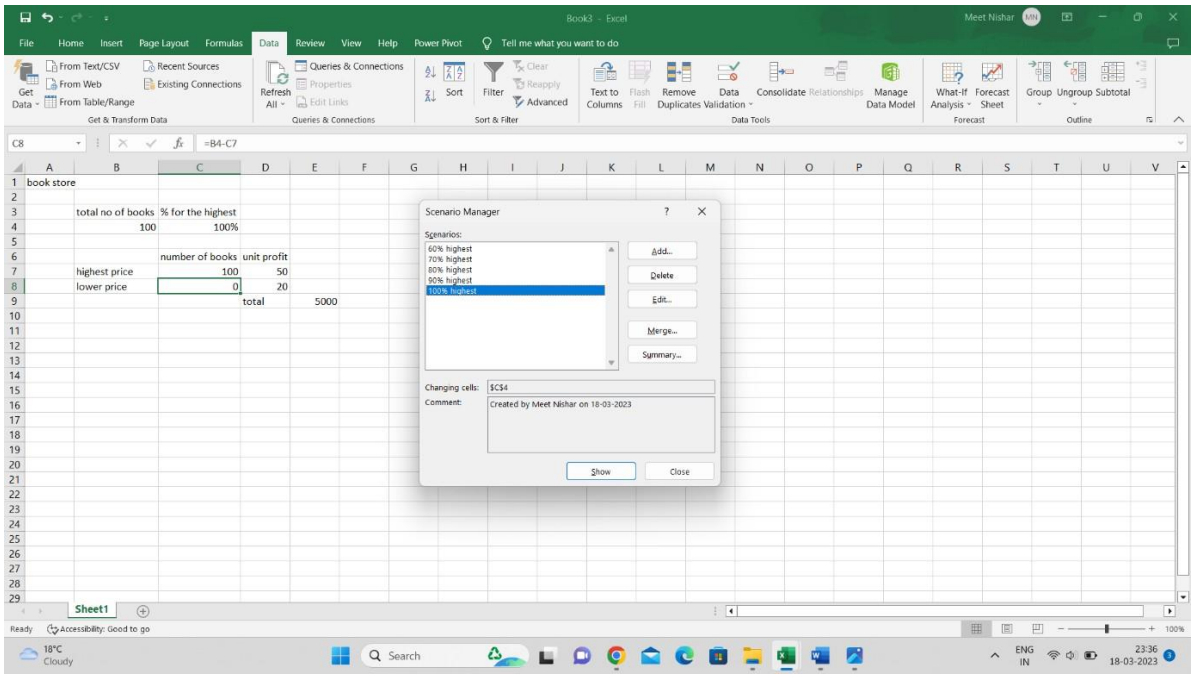
7. Type one more name (70% highest), select cell C4 (% sold for the highest price) for the Changing cells and Click on OK.
8. Enter the corresponding value 0.7 and click on OK.



	A	B	C	D	E	F
1	book store					
2						
3		total no of books	% for the highest			
4		100	70%			
5						
6			number of books	unit profit		
7		highest price	70	50		
8		lower price	30	20		
9				total	4100	
10						
11						
12						

9. Next, add 3 other scenarios (80%, 90%, 100%)

Finally, your Scenario Manager should be consistent with the picture below:


Book3 - Excel
Meet Nishar
File Home Insert Page Layout Formulas Data Review View Help Power Pivot Tell me what you want to do
Get Data From Text/CSV From Web From Table/Range Recent Sources Existing Connections
Refresh All Queries & Connections Properties Edit Links
Sort Filter Clear Reapply Advanced
Text to Columns Flash Fill Remove Duplicates Data Validation Consolidate Relationships Manage Data Model
What-If Analysis Forecast Sheet
Group Ungroup Subtotal
Get & Transform Data
Queries & Connections
Sort & Filter
Data Tools
Forecast
Outline
C8
=B4-C7
book store
total no of books
% for the highest
100
100%
number of books
unit profit
highest price
100
50
lower price
0
20
total
5000
Scenario Manager
Scenarios:
60% highest
70% highest
80% highest
90% highest
100% highest
Add...
Delete
Edit...
Merge...
Summary...
Changing cells: C4
Comment:
Created by Meet Nishar on 18-03-2023
Show
Close
Sheet1
Ready
Accessibility: Good to go
18°C Cloudy
Search
ENG IN
23:36 18-03-2023

Practical 8

Data Analyzing using Time-series Analysis.

- Time series is a series of data points in which each data point is associated with a timestamp. A simple example is the price of a stock in the stock market at different points of time on a given day. Another example is the amount of rainfall in a region at different months of the year. R language uses many functions to create, manipulate and plot the time series data. The data for the time series is stored in an R object called time-series object. It is also a R data object like a vector or data frame.

- The time series object is created by using the ts() function.

- The basic syntax for ts() function in time series analysis is −

timeseries.object.name <- ts(data, start, end, frequency)

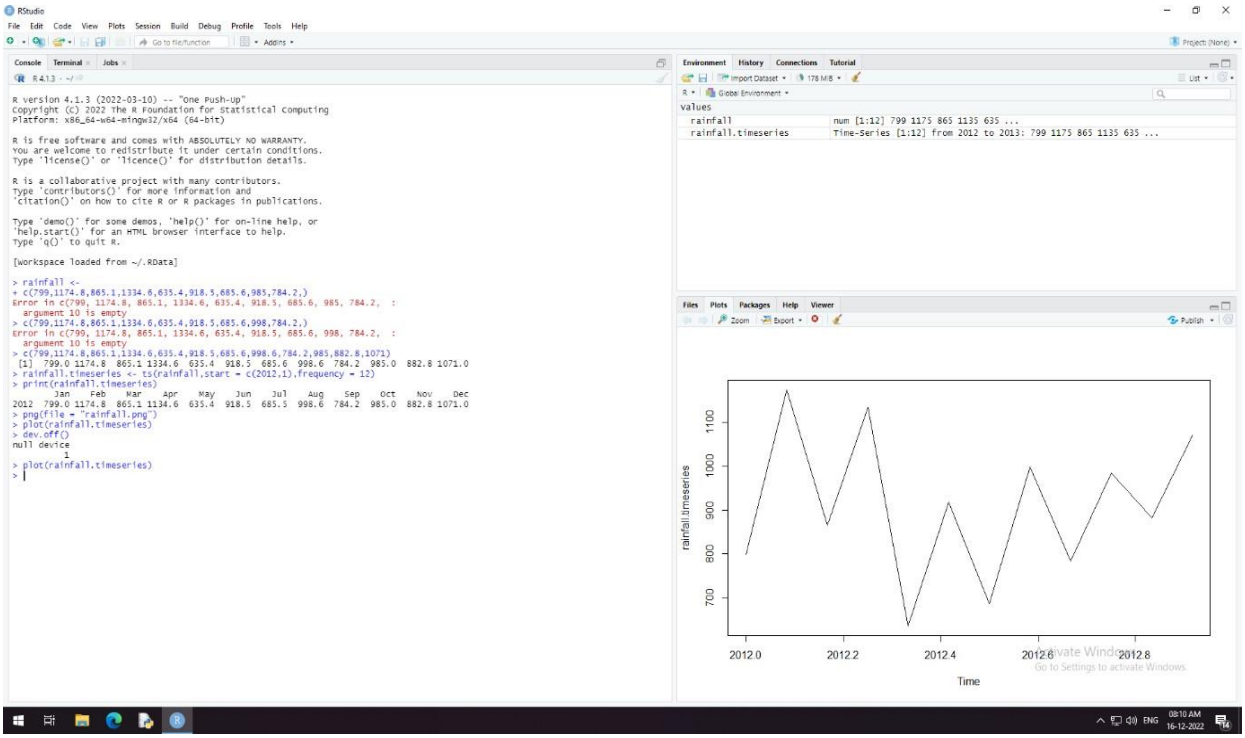
- Following is the description of the parameters used −
  - Data is a vector or matrix containing the values used in the time series.
  - Start specifies the start time for the first observation in time series.
  - End specifies the end time for the last observation in time series.
  - Frequency specifies the number of observations per unit time.

Except the parameter "data" all other parameters are optional.

- Consider the annual rainfall details at a place starting from January 2012. We create an R time series object for a period of 12 months and plot it.

```
# Get the data points in form of a R vector.
rainfall <-c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

# Convert it to a time series object.
rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)

# Print the timeseries data.
print(rainfall.timeseries)

# Give the chart file a name.
png(file = "rainfall.png")

# Plot a graph of the time series.
plot(rainfall.timeseries)

# Save the file.
dev.off()
```

Output


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Jobs
R 4.1.3 · ~/
R version 4.1.3 (2022-03-10) -- "One Push-up"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
> rainfall <-
+ c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,985,784.2,)
Error in c(799, 1174.8, 865.1, 1334.6, 635.4, 918.5, 685.6, 985, 784.2, :
argument 10 is empty
> c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,998,784.2,)
Error in c(799, 1174.8, 865.1, 1334.6, 635.4, 918.5, 685.6, 998, 784.2, :
argument 10 is empty
> c(799,1174.8,865.1,1334.6,635.4,918.5,685.6,998.6,784.2,985,882.8,1071)
[1] 799.0 1174.8 865.1 1334.6 635.4 918.5 685.6 998.6 784.2 985.0 882.8 1071.0
> rainfall.timeseries <- ts(rainfall,start = c(2012,1),frequency = 12)
> print(rainfall.timeseries)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2012 799.0 1174.8 865.1 1334.6 635.4 918.5 685.5 998.6 784.2 985.0 882.8 1071.0
> png(file = "rainfall.png")
> plot(rainfall.timeseries)
> dev.off()
null device
1
> plot(rainfall.timeseries)
Environment History Connections Tutorial
Import Dataset 178 MiB
Global Environment
values
rainfall num [1:12] 799 1175 865 1135 635 ...
rainfall.timeseries Time-Series [1:12] from 2012 to 2013: 799 1175 865 1135 635 ...
Files Plots Packages Help Viewer
Zoom Export Publish
rainfall.timeseries
700
800
900
1000
1100
2012.0
2012.2
2012.4
2012.6
2012.8
Time
Activate Windows
Go to Settings to activate Windows.
ENG
08:10 AM
16-12-2022

Practical 9

Implementation of Decision Tree using R Tool.

Install.packages("party") - The package "party" has the function ctree() which is used to create and analyze decison tree

Syntax:

The basic syntax for creating a decision tree in R is − ctree(formula, data)

Input Data:

We will use the R in-built data set named readingSkills to create a decision tree. It describes the score of someone's readingSkills if we know the variables "age","shoesize","score" and whether the person is a native speaker or not.

```
# Load the party package. It will automatically load other dependent packages.

library(party)

# Print some records from data set readingSkills.

print(head(readingSkills))
```

We will use the ctree() function to create the decision tree and see its graph.

```
# Load the party package. It will automatically load other dependent packages.

library(party)

# Create the input data frame.

input.dat <- readingSkills[c(1:105),]
```

```
# Give the chart file a name.
png(file = "decision_tree.png")
# Create the tree.
 output.tree <- ctree(nativeSpeaker ~ age + shoeSize + score, data = input.dat)
# Plot the tree.
plot(output.tree)
# Save the file.
dev.off
```

Output: