ASSIGNMENT No: 07

Title: Import Data from different Sources such as (Excel, Sql Server, Oracle etc.) and load in targeted system.

Problem Statement: Import Data from different Sources such as (Excel, Sql Server, Oracle etc.) and load in targeted system.

Prerequisite:

Basics of Python

Software Requirements: Power BI Tool

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to import data from different sources such as(Excel, Sql Server, Oracle etc.) and load in targeted system.

Outcomes:

After completion of this assignment students are able to understand how to import data from different sources such as(Excel, Sql Server, Oracle etc.) and load in targeted system.

Theory:

What is Legacy Data?

Legacy data, according to Business Dictionary, is "information maintained in an old or outof-date format or computer system that is consequently challenging to access or handle."

Sources of Legacy Data

Where does legacy data come from? Virtually everywhere. Figure 1 indicates that there are many sources from which you may obtain legacy data. This includes existing databases, often relational, although non-RDBs such as hierarchical, network, object, XML, object/relational databases, and NoSQL databases. Files, such as XML documents or "flat files"ù such as configuration files and comma-delimited text files, are also common sources of legacy data. Software, including legacy applications that have been wrapped (perhaps via CORBA) and legacy services such as web services or CICS transactions, can also provide

access to existing information. The point to be made is that there is often far more to gaining access to legacy data than simply writing an SQL query against an existing relational database.

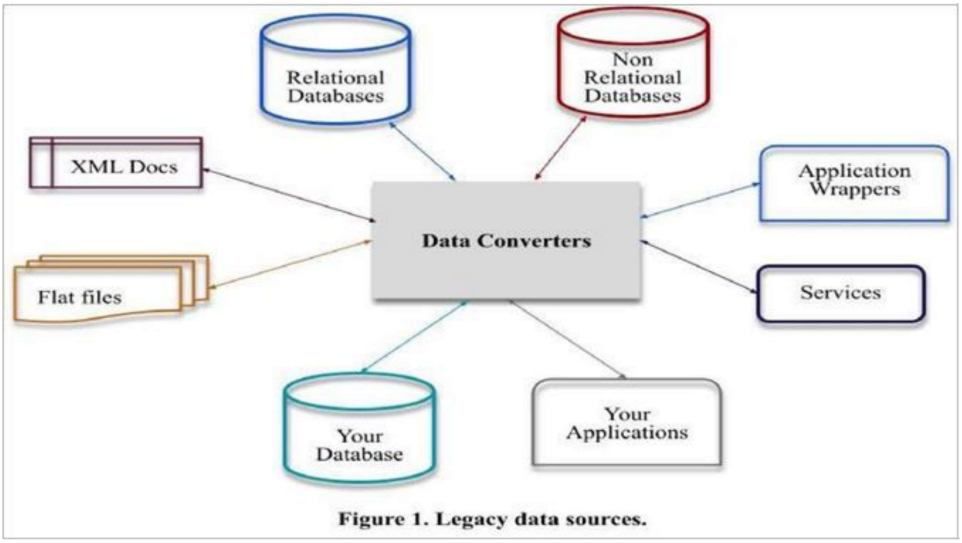
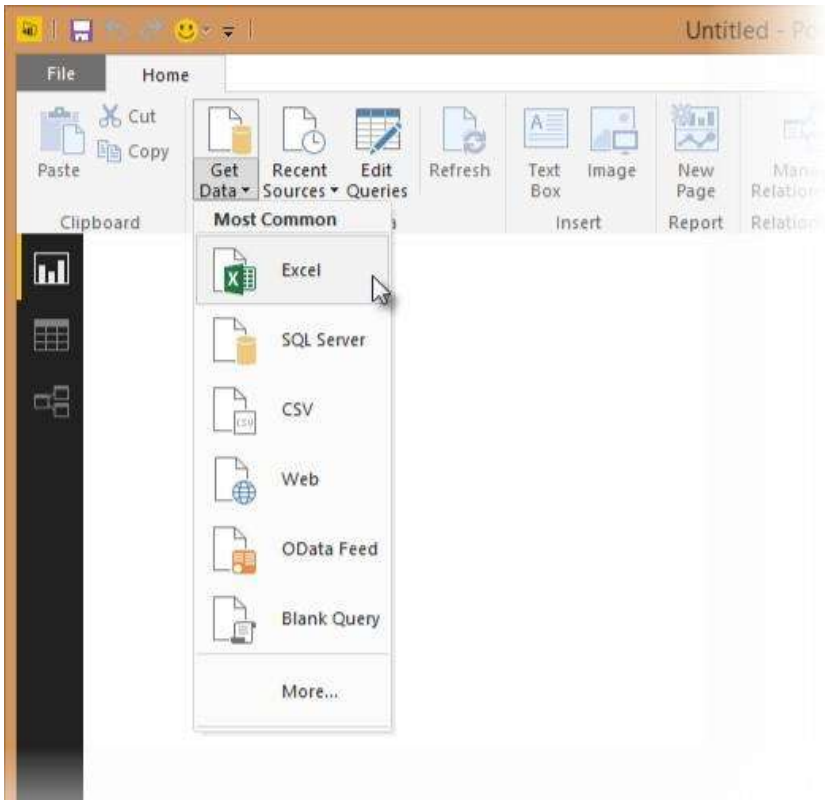


Figure 1. Legacy data sources.

Importing Excel Data

1) Launch Power BI Desktop.

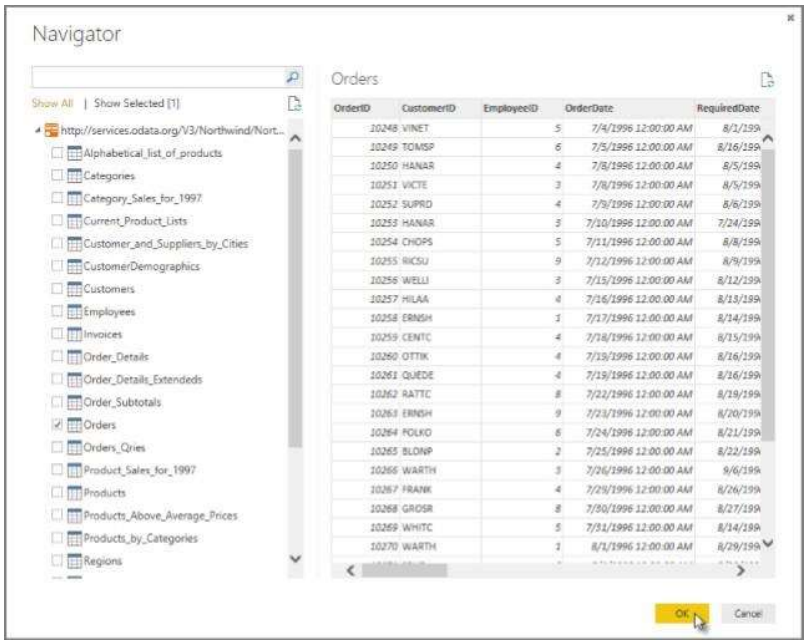
2) From the Home ribbon, select Get Data. Excel is one of the Most Common data connections, so you can select it directly from the Get Data menu.



3) If you select the Get Data button directly, you can also select FIle > Excel and select Connect.

4) In the Open File dialog box, select the Products.xlsx file.

5) In the Navigator pane, select the Products table and then select Edit.



Conclusion: - This way, Implemented a program for inverted files.

ASSIGNMENT No: 08

Title: Data Visualization from Extraction Transformation and Loading (ETL) Process.

Problem Statement: Data Visualization from Extraction Transformation and Loading (ETL) Process.

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn Data Visualization from Extraction Transformation and Loading (ETL) Process

Outcomes:

After completion of this assignment students are able to understand how Data Visualization is done through Extraction Transformation and Loading (ETL) Process

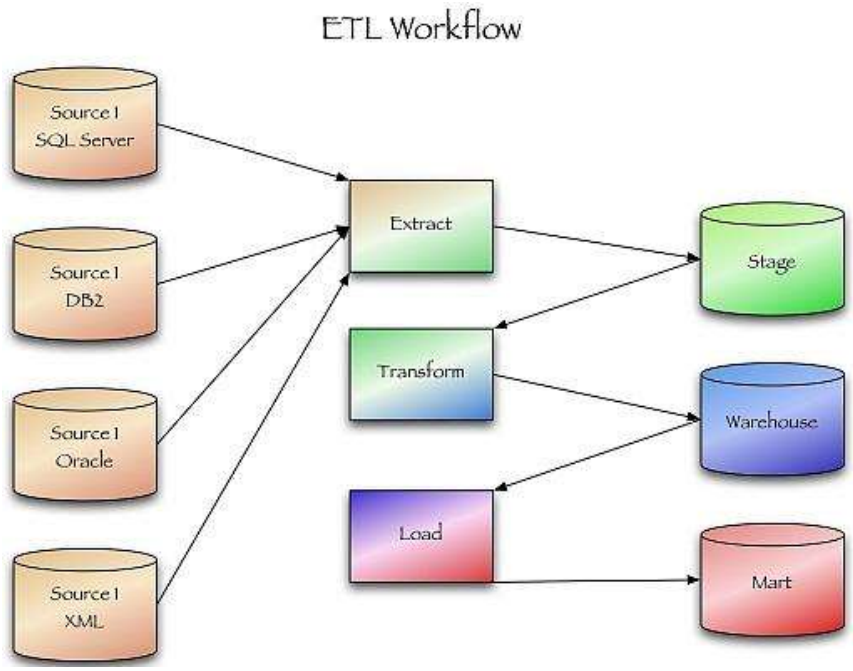
Theory:

Extract, transform, and load (ETL) are 3 data processes, followed after data collection. Extraction takes data, collected in data sources like flat files, databases (relational, hierarchical etc.), transactional datastores, semi-structured repositories (e.g. email systems or document libraries) with different structure and format, pre-validating extracted data and parsing valid data to destination (e.g. staging database)

Transformation takes extracted data and applies predefined rules and functions to it, including selection (e.g. ignore or remove NULLs), data cleansing, encoding (e.g. mapping “Male” to “M”), deriving (e.g. calculating designated value as a product of extracted value and predefined constant) , sorting, joining data from multiple sources (e.g. lookup or merge), aggregation (e.g. summary for each month), transposing (columns to rows or vice versa), splitting, disaggregation, lookups (e.g. validation through dictionaries), predefined validation etc. which may lead to rejection of some data. Transformed data can be stored into Data Warehouse (DW).

Load takes transformed data and places it into end target, in most cases called Data Mart (sometimes they called Data Warehouse too). Load can append, refresh or/and overwrite preexisting data, apply constraints

and execute appropriate triggers (to enforce data integrity, uniqueness, mandatory fields, provide log etc.) and may start additional processes, like data backup or replication.



Conclusion:- This way Data Visualization from Extraction Transformation and Loading (ETL) Process is done.

ASSIGNMENT No: 09

Title: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.

Problem Statement: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.

Outcomes:

After completion of this assignment students are able to understand how to Perform the Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.

Theory:

Step 1 : Data Extraction :

The data extraction is first step of ETL. There are 2 Types of Data Extraction

1. Full Extraction : All the data from source systems or operational systems gets extracted to staging area. (Initial Load)

2. Partial Extraction : Sometimes we get notification from the source system to update specific date. It is called as Delta load.

Source System Performance: The Extraction strategies should not affect source system performance.

Step 2 : Data Transformation :

The data transformation is second step.After extracting the data there is big need to do the transformation as per the target system.I would like to give you some bullet points of Data Transformation.

- Data Extracted from source system is in to Raw format. We need to transform it before loading in to target server.
- Data has to be cleaned, mapped and transformed.
- There are following important steps of Data Transformation :

1.Selection : Select data to load in target

2.Matching : Match the data with target system

3.Data Transforming : We need to change data as per target table structures

Real life examples of Data Transformation :

- Standardizing data : Data is fetched from multiple sources so it needs to be standardized as per the target system.
- Character set conversion : Need to transform the character sets as per the target systems. (Firstname and last name example)
- Calculated and derived values: In source system there is first val and second val and in target we need the calculation of first val and second val.
- Data Conversion in different formats : If in source system date in in DDMMYY format and in target the date is in DDMONYYYY format then this transformation needs to be done at transformation phase.

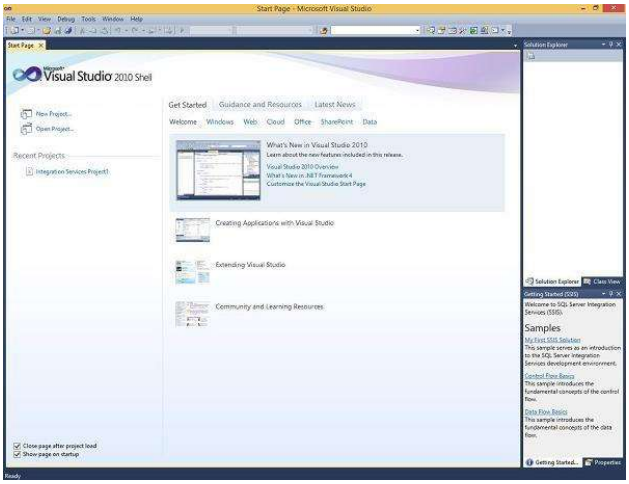
Step 3 : Data Loading

- Data loading phase loads the prepared data from staging tables to main tables.

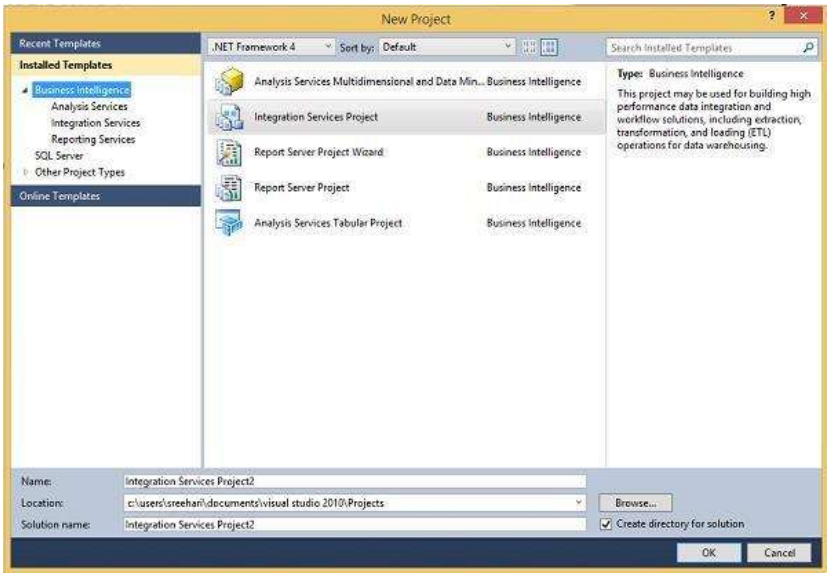
ETL process in SQL Server:

Following are the steps to open BIDS\SSDT.

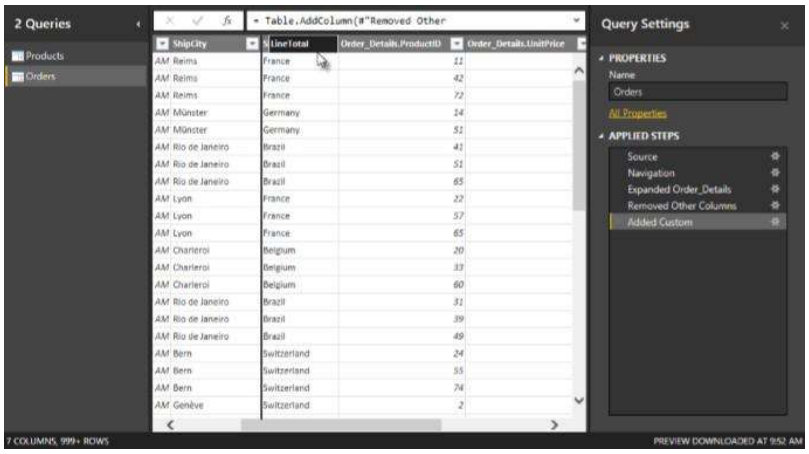
Step 1 − Open either BIDS\SSDT based on the version from the Microsoft SQL Server programs group. The following screen appears.



Step 2 − The above screen shows SSDT has opened. Go to file at the top left corner in the above image and click New. Select project and the following screen opens.



Step 3 − Select Integration Services under Business Intelligence on the top left corner in the above screen to get the following screen.



2.Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.

6. Combine the Products and Total Sales queries

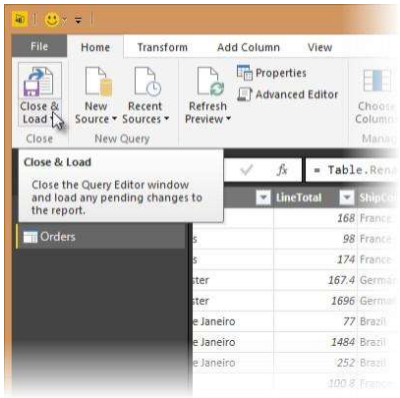
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets

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.

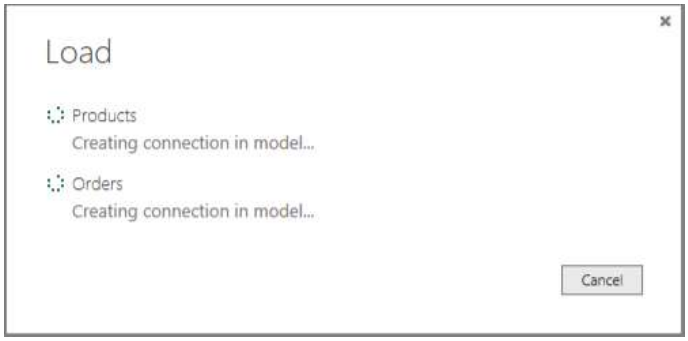
In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries

Step 1: Confirm the relationship between Products and Total Sales

1. First, we need to load the model that we created in Query Editor into Power BI Desktop. From the Home ribbon of Query Editor, select Close & Load.

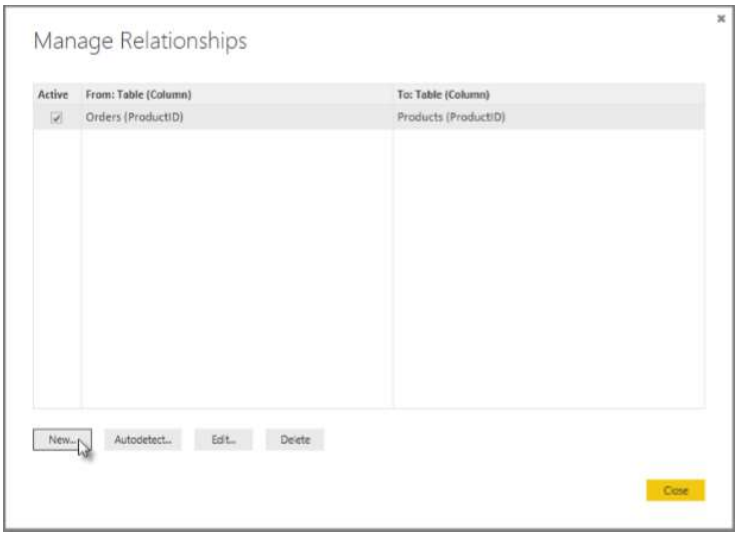


1. Power BI Desktop loads the data from the two queries.

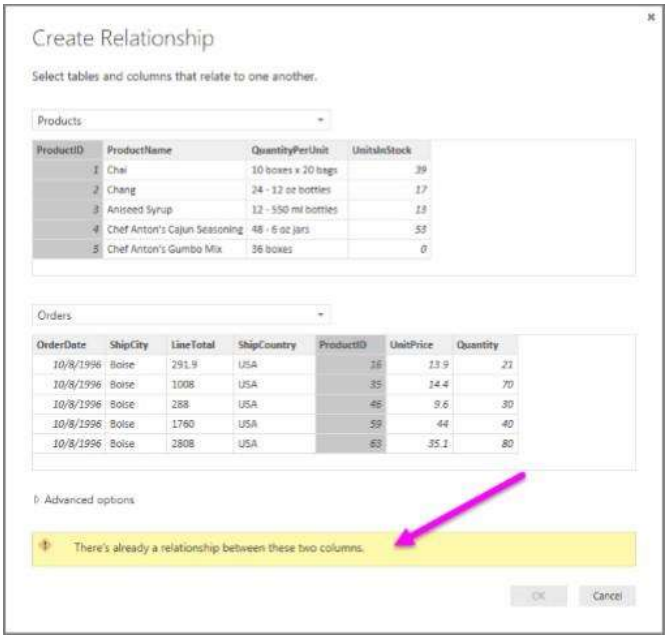


2. Once the data is loaded, select the Manage Relationships button Home ribbon.

3. Select the New… button

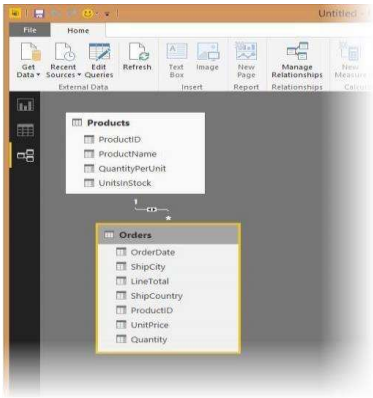


4. When we attempt to create the relationship, we see that one already exists! As shown in theCreate Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.

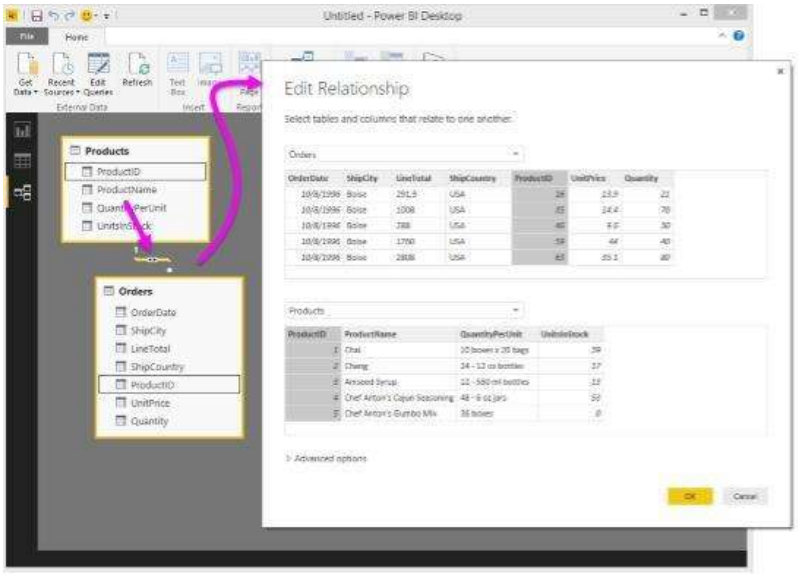


5. Select Cancel, and then select Relationship view in Power BI Desktop.

6. We see the following, which visualizes the relationship between the queries.



7. When you double-click the arrow on the line that connects the to queries, an EditRelationship dialog appears.



No need to make any changes, so we'll just select Cancel to close the EditRelationship dialog

Conclusion: Thus Performed Extraction Transformation and Loading (ETL) process to construct the database in the Sql server / Power BI.

ASSIGNMENT No: 10

Title: Data Analysis and Visualization using Advanced Excel.

Problem Statement: Data Analysis and Visualization using Advanced Excel.

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to Perform Data Analysis and Visualization using Advanced Excel.

Outcomes:

After completion of this assignment students are able to understand Data Analysis and Visualization using Advanced Excel.

Theory:

Defining Data Visualization

We'll first start by defining what data visualization is. Data visualization is a graphical representation of data. By utilizing charts, graphs, maps, etc., we can provide a simple and accessible way to understand our data and identify trends and outliers within our datasets. Note that Excel uses the term "chart" to mean a "plot". For example, a bar plot is called a bar chart in Excel terminology.

The purpose of this tutorial is to walk you through some basic charts to visualize your data before jumping into more advanced techniques later on. We highly recommend you check out our Data Visualization cheat sheet to learn more about the most common visualizations and when to use them.

Example Dataset

We first need a dataset to work with before creating any visualizations. This tutorial will use a simple dataset containing sales data for a local electronics store. The dataset contains information on the number of units sold for various product types in 2022 and totals for columns and rows.

Month	TVs	Mobile Phones	Laptops	Total
1/1/2022	145	335	82	562
2/1/2022	145	362	126	633
3/1/2022	105	311	95	511
4/1/2022	171	259	93	523
5/1/2022	178	277	107	562
6/1/2022	167	292	145	604
7/1/2022	200	385	77	662
8/1/2022	181	388	78	647

9/1/2022	152	291	83	526
10/1/2022	143	345	102	590
11/1/2022	114	399	99	612
12/1/2022	109	250	101	460
Total	1810	3894	1188	

We'll be working with this dataset throughout this tutorial. You can download the data file from GitHub.

Alternatively, you can import the dataset using the following steps:

- Open Excel and create a new workbook.
- Copy the dataset above and paste it into cell A1

Format the cells as needed (e.g., adjust column width, apply bold formatting to headers, etc.).

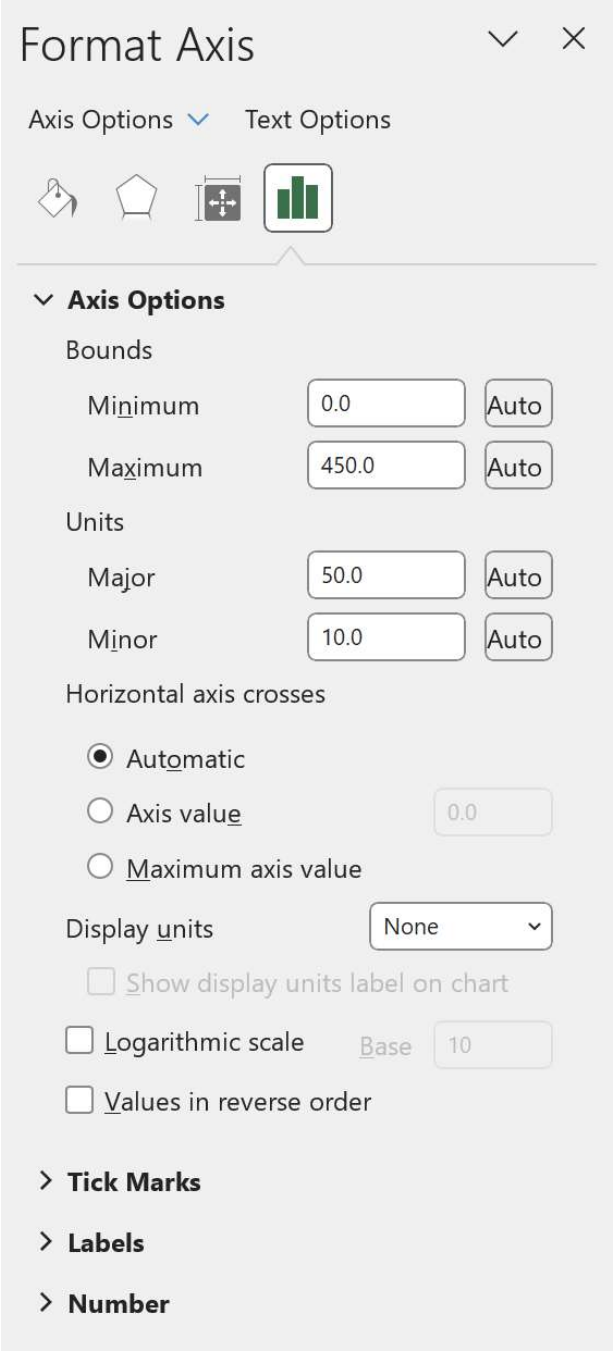
	A	B	C	D
1	**Month**	**Television**	**Laptop**	**Mobile Phones**
2	1/1/2022	145	335	82
3	2/1/2022	145	362	126
4	3/1/2022	105	311	95
5	4/1/2022	171	259	93
6	5/1/2022	178	277	107
7	6/1/2022	167	292	145
8	7/1/2022	200	385	77
9	8/1/2022	181	388	78
10	9/1/2022	152	291	83
11	10/1/2022	143	345	102
12	11/1/2022	114	399	99
13	12/1/2022	109	250	101

Creating Basic Charts in Excel

Excel has multiple options for choosing a particular chart type. For example, if you want to create a column or bar chart, you are often presented with various visualization options. For example, there are 2D and 3D versions and normal, stacked, and 100% stacked options. Depending on your requirements, you can choose the visualization type that best suits your needs.

- Right-click the axis you want to modify and choose "Format Axis"

In the "Format Axis" pane, you can change the minimum and maximum values, major and minor units, or number format.



For this tutorial, we intend to keep these options the same but feel free to play around and test what each does.

Axis visibility

Finally, if you would like to remove the axis labels from showing, you can take the following steps:

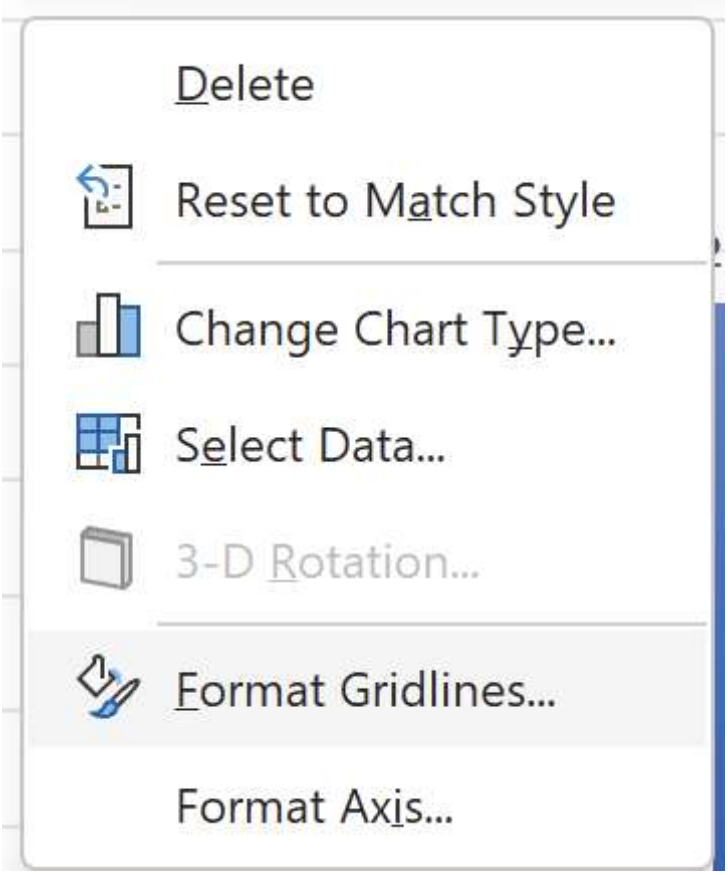
- In the Excel ribbon, click the "Chart Design" tab
- Click the "Add Chart Element" dropdown and navigate to "Axes"
- By default, both options will have a darker gray box surrounding them, to remove an Axes, simply de-select the axes you'd like to remove

Other Excel formatting options

There are other formatting options available, including modifying data series colors, adjusting chart and plot area backgrounds, and customizing gridlines.

To access these options:

- Right-click the chart element you want to modify (e.g., data series, plot area, or gridlines)
- Select "Format [element]"



Conclusion:- Thus, this way Data Analysis and Visualization is done using Advanced Excel.

ASSIGNMENT No: 11

Title: Perform the data classification algorithm using any Classification algorithm

Problem Statement: Perform the data classification algorithm using any Classification algorithm

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to Perform the data classification algorithm using any Classification algorithm

Outcomes:

After completion of this assignment students are able to understand Perform the data classification algorithm using any Classification algorithm

Theory:

What is Classification?

We use the training dataset to get better boundary conditions which could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class. The whole process is known as classification.

Target class examples:

- Analysis of the customer data to predict whether he will buy computer accessories (Target class: Yes or No)
- Classifying fruits from features like color, taste, size, weight (Target classes: Apple, Orange, Cherry, Banana)
- Gender classification from hair length (Target classes: Male or Female)

Let's understand the concept of classification algorithms with gender classification using hair length (by no means am I trying to stereotype by gender, this is only an example). To classify gender (target class) using hair length as feature parameter we could train a model using any classification algorithms to come up with some set of boundary conditions which can be used to differentiate the male and female genders using hair length as the training feature. In gender classification case the boundary condition could the proper hair length value. Suppose the differentiated boundary hair length value is 15.0 cm then we can say that if hair length is less than 15.0 cm then gender could be male or else female.

Classification Algorithms vs Clustering Algorithms

In clustering, the idea is not to predict the target class as in classification, it's more ever trying to group the similar kind of things by considering the most satisfied condition, all the items in the same group should be similar and no two different group items should not be similar.

Group items Examples:

- While grouping similar language type documents (Same language documents are one group.)
- While categorizing the news articles (Same news category(Sport) articles are one group)

Let's understand the concept with clustering genders based on hair length example. To determine gender, different similarity measure could be used to categorize male and female genders. This could be done by finding the similarity between two hair lengths and keep them in the same group if the similarity is less (Difference of hair length is less). The same process could continue until all the hair length properly grouped into two categories.

Basic Terminology in Classification Algorithms

- Classifier: An algorithm that maps the input data to a specific category.
- Classification model: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- Feature: A feature is an individual measurable property of a phenomenon being observed.
- Binary Classification: Classification task with two possible outcomes. Eg: Gender classification (Male / Female)
- Multi-class classification: Classification with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. Eg: An animal can be a cat or dog but not both at the same time.
- Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drugs classification
- Facial key points detection
- Pedestrians detection in an automotive car driving.

Types of Classification Algorithms

Classification Algorithms could be broadly classified as the following:

- Linear Classifiers
  - Logistic regression
  - Naive Bayes classifier
  - Fisher's linear discriminant
- Support vector machines
  - Least squares support vector machines
- Quadratic classifiers
- Kernel estimation
  - k-nearest neighbor
- Decision trees
  - Random forests
- Neural networks
- Learning vector quantization

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.

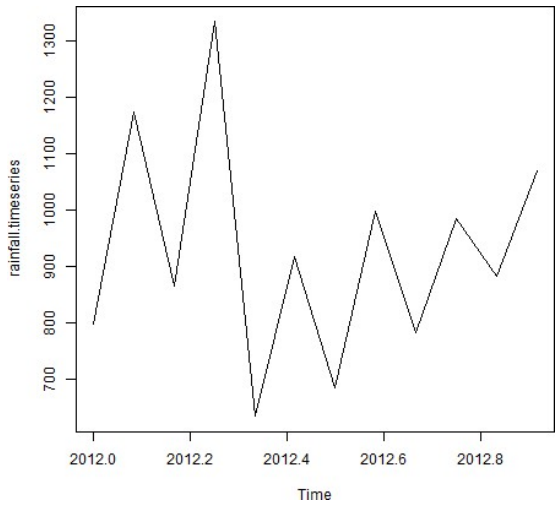
```
# Get the data points in form of a R vector.rainfall <-
c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

# 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:

When we execute the above code, it produces the following result and chart −

```
Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep
2012    799.0    1174.8    865.1    1334.6    635.4    918.5    685.5    998.6    784.2
            Oct       Nov       Dec
2012    985.0    882.8 1071.0
```



Conclusion:- Thus, this way Performed data classification algorithm using any Classification algorithm

ASSIGNMENT No: 12

Title: Perform the data clustering algorithm using any Clustering algorithm

Problem Statement: Implement Page Rank Algorithm. (Use python or beautiful soup for implementation).

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to Perform the data clustering algorithm using any Clustering algorithm

Outcomes:

After completion of this assignment students are able to understand how to Perform the data clustering algorithm using any Clustering algorithm

Theory:

Clustering your data can provide a new way to slice that is based on the properties of the data instead of other labels. For instance, customer data is often sliced by demographic parameters like gender, age, location, etc. This data can be useful in many cases, but what if you could slice your customers by their behaviour? What they buy, how often, how much they spend, etc. This information can help with advertising because you are now looking at past behaviour that can correlate better with future actions than demographics.

k-Mean Clustering

From the results of my testing, I believe the algorithm responsible for clustering in Power BI is the k-means algorithm. I did not find any confirmation on this, but it seems reasonable given the results found below. Knowing this can help you understand how Power BI finds clusters and how it will work in the situation you are using.

The goal of k-means is to minimize the distance between the points of each cluster. Each cluster has a centre. Data points are labeled as part of a cluster depending on which centre they are closest to.

As a result, certain types of clusters are easy to find, and in others, the algorithm will fail. Below, you will see examples of both cases.

```
R Console

> # Apply K mean to iris and store result
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris,3))
K-means clustering with 3 clusters of sizes 62, 38, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     6.850000    3.073684     5.742105    2.071053
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [47] 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [93] 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2
[139] 1 2 2 2 1 2 2 2 1 2 2 1

Within cluster sum of squares by cluster:
[1] 39.82097 23.87947 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> |
```

Compare the Species label with the clustering result

```
R Console

> #Compare the Species label with the clustering result
> table (iris$Species,kc$cluster)

               1  2  3
  setosa       0  0 50
  versicolor  48  2  0
  virginica   14 36  0
> |
```

Plot the clusters and their centre

Conclusion:- Thus, this way Performed data clustering algorithm using any Clustering algorithm