Sardar Patel Institute of Technology,Mumbai
Department of Electronics and Telecommunication Engineering
T.E. Sem-V (2018-2019)
ETL54-Statistical Computational Laboratory
Lab-8:Data Visualization and Matrix Computation

Name:Manish D'Silva Roll No.15 Part-I: Data Visualization

Objective:To create a range of graphs to summarize your data and results

Outcomes:

1. To create boxplot, scatter plots, including correlation plots
2. To create line graphs, pie charts and bar charts
3. To save graphs as files on disk (png, jpg etc)
4. To choose the right type of chart for your specific objectives and how to implement it in R using ggplot2.

System Requirements: Ubuntu OS with R and RStudio installed and ggplot2,Python, Pandas, Matplotlib,seaborn, Plotly etc.

Introduction to Visualization:

Data visualization is an art of how to turn numbers into useful knowledge. [1] Graphs are a powerful way to present your data and results in a concise manner. Whatever kind of data you have, there is a way to illustrate it graphically. A graph is more readily understandable than words and numbers, and producing good graphs is a vital skill. Some graphs are also useful in examining data so that you can gain some idea of patterns that may exist; this can direct you toward the correct statistical analysis.

Selecting the Right Chart Type:

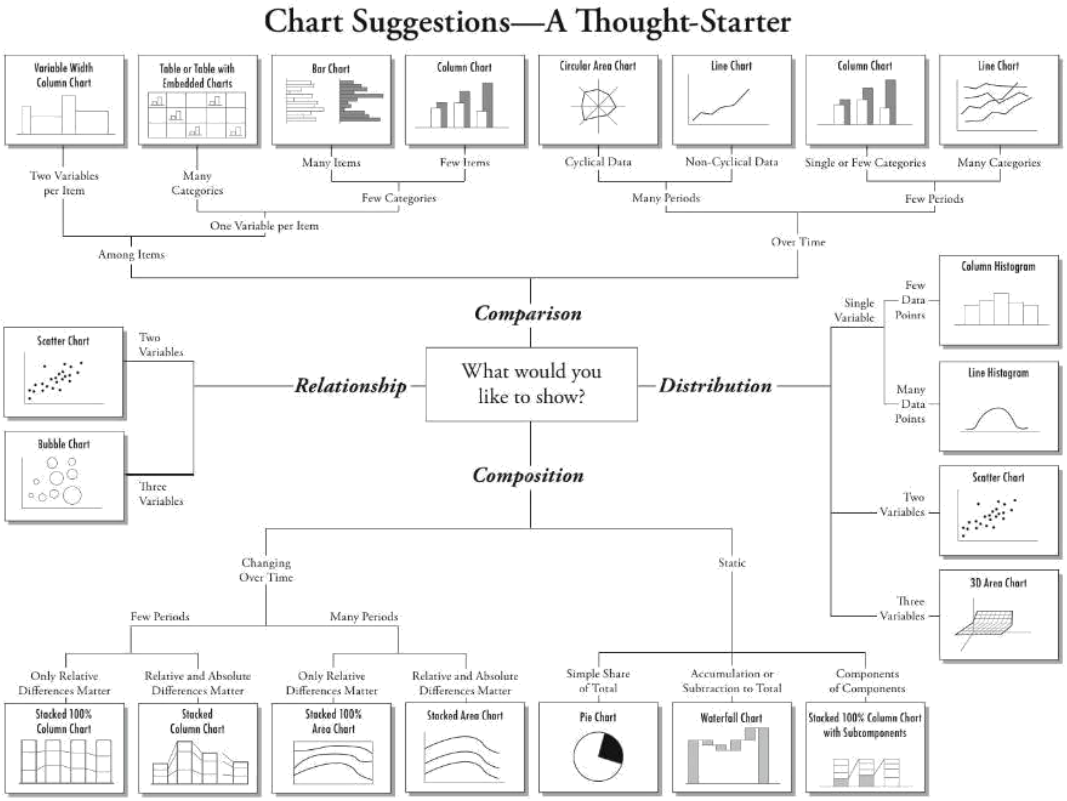
There are four basic presentation types:

1. Comparison
2. Composition
3. Distribution
4. Relationship

To determine which amongst these is best suited for your data, We suggest you should answer a few questions like,[1]

- How many variables do you want to show in a single chart?
- How many data points will you display for each variable?
- Will you display values over a period of time, or among items or groups?

Do refer the following picture and understand how select a right chart type.



[Courtesy: Dr. Andrew Abela]

In your day-to-day activities, you'll come across the below listed 7 charts most of the time.

1. Scatter Plot
2. Histogram
3. Bar & Stack Bar Chart
4. Box Plot
5. Area Chart
6. Heat Map
7. Correlogram

[1] Introduction to Data Visualization in Python- [1 hr]

https://towardsdatascience.com/introduction-to-data-visualization-in-python-89a54c97fbed

Perform this lab using the KDDCUP99 Intrusion Dataset
Refer the Lab6: Anomaly Detection using Machine Learning
Download the datasets (CSVs): i. Train ii. Test Additional:

[2] **Pre-reading material and understand (for 60 minutes)** Refer the following website; https://www.r-bloggers.com/7-visualizations-you-should-learn-in-r/

[3] Read and perform laboratory on data visualization with R Graphics [2 hr]

Refer the following website: **http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html Laboratory Session:**

Describe following with respect to data visualization:

1. When to use scatter plot, histogram,bar and stack charts,box plot, Area chart?

- **When to Use . . .**

  . . . a Line graph.

  Line graphs are used to track changes over short and long periods of time. When smaller changes exist, line graphs are better to use than bar graphs. Line graphs can also be used to compare changes over the same period of time for more than one group.

  . . . a Pie Chart.

  Pie charts are best to use when you are trying to compare parts of a whole. They do not show changes over time.

  . . . a Bar Graph.

  Bar graphs are used to compare things between different groups or to track changes over time. However, when trying to measure change over time, bar graphs are best when the changes are larger.

  . . . an Area Graph.

  Area graphs are very similar to line graphs. They can be used to track changes over time for one or more groups. Area graphs are good to use when you are tracking the changes in two or more related groups that make up one whole category (for example public and private groups).

  . . . an X-Y Plot.

  X-Y plots are used to determine relationships between the two different things. The x-axis is used to measure one event (or variable) and the y-axis is used to measure the other. If both variables increase at the same time, they have a positive relationship. If one variable decreases while the other increases, they have a negative relationship. Sometimes the variables don't follow any pattern and have no relationship.

2. What is Heatmap?

- A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets.
- There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be would be much harder to understand if presented numerically in a spreadsheet.
- In the United States, many people are familiar with heat maps from viewing television news programs. During a presidential election, for instance, a geographic heat map with the colors red and blue will quickly inform the viewer which states each candidate has won

3. What is correlogram?

- A correlogram (also called Auto Correlation Function ACF Plot or Autocorrelation plot) is a visual way to show serial correlationin data that changes over time (i.e. time

seriesdata). Serial correlation (also called autocorrelation) is where an error at one point in time travels to a subsequent point in time. For example, you might overestimate the value of your stock market investments for the first quarter, leading to an overestimate of values for following quarters.

- Correlograms can give you a good idea of whether or not pairs of data show autocorrelation. They cannot be used for measuring how large that autocorrelation is (for a mathematical way to test for serial correlation, try the Durbin Watson test).

4. How to summarize lots of data?

There are four key areas to consider when summarizing a set of numbers:

- Centrality– the middle value or average.
- Dispersion– how spread out the values are from the average.
- Replication– how many values there are in the sample.
- Shape– the data distribution, which relates to how "evenly" the values are spread either side of the average.

Conclusion:

- We have plotted a correlation plot between features of our dataset. It is a heatmap of colour code according to Pearson Correlation Coefficient
- We did this to summarize data and know the relationship among features and between features and target variables
- It would be helpful for Analyzing Data for business insights and explain your findings with the help of Graphs and Visuals

References:

[1] **7 Visualizations You Should Learn in R**

https://www.r-bloggers.com/7-visualizations-you-should-learn-in-r/

[2] **Top 50 ggplot2 Visualizations**
http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

Part-II: Matrix Computation

Objective: To carry out matrix computation

Outcomes:

5. To create vectors and matrices
6. To extract elements from a matrix
7. To use and describe the general information commands with respect to matrix.
8. To carry out matrix operations
9. To find the eigenvalues and eigenvectors

System Requirements: Ubuntu OS with R and RStudio installed, Python etc

Introduction to Linear Algebra:

Procedure:

1. Refer the following for Matrix Computation in Python

Matrix Arithmetics under NumPy and Python
https://www.python-course.eu/matrix_arithmetic.php

2. Gentle Introduction to Eigenvalues and Eigenvectors for Machine Learning
https://machinelearningmastery.com/introduction-to-eigendecomposition-eigenvalues-and-eigenvectors/

[3] Refer the [1] and [2] pdf files provided and complete the lab.

Describe the following with respect to matrix computation

- List the general information commands used in Python for matrix

ndarray.ndim

the number of axes (dimensions) of the array.

ndarray.shape

the dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with *n* rows and *m* columns, shape will be (n,m). The length of the shapetuple is therefore the number of axes, ndim.

ndarray.size

the total number of elements of the array. This is equal to the product of the elements of shape.

ndarray.dtype

an object describing the type of the elements in the array. One can create or specify dtype's using standard Python types. Additionally NumPy provides types of its own. numpy.int32, numpy.int16, and numpy.float64 are some examples.

ndarray.itemsize

the size in bytes of each element of the array. For example, an array of elements of type float64 has itemsize 8 (=64/8), while one of type complex32 has itemsize 4 (=32/8). It is equivalent to ndarray.dtype.itemsize.

ndarray.data

the buffer containing the actual elements of the array. Normally, we won't need to use this attribute because we will access the elements in an array using indexing facilities.

- Describe the importance of matrix computation.
  Graph Theory --loosely, the study of connect-the-dot figures-- uses matrices to encode adjacency and incidence structures. More than simply bookkeeping, however, the matrices have computational uses. From powers of the adjacency matrix, for a simple example, one can read the number of available paths between any two dots.

  "Spectral" Graph Theory derives graph-theoretical information from matrix-theoretical results (specifically, "eigenvalues" and "eigenvectors" --by the way, the set of eigenvalues is the "spectrum" of a matrix, hence "spectral"--which come from the linear map interpretation of matrices).geometric realizations of graphs --think Platonic and Archimedean solids-- from this kind of analysis of their adjacency matrices.

  For all this matrix needs to be computed

Conclusion:

- We have created basic Vectors and Matrices and realized basic computations of linear algebra
- We have used Vectorized Math techniques which are essential for fast computation for various algorithms
- We have calculated Eigen Values and Eigen Vectors which is used in PCA.I t is a technique to reduce the dimensionality of the data and helps visualize the features using a 2D or 3D plot

References:

[1]Examples of Using R with Linear Algebra by S. K. Hyde [pdf]
[2]Linear algebra in R by Søren Højsgaard [pdf]
[3]https://www.statmethods.net/advstats/matrix.html
[4] Hands-On Matrix Algebra Using R by Hrishikesh D Vinod, World Scientific