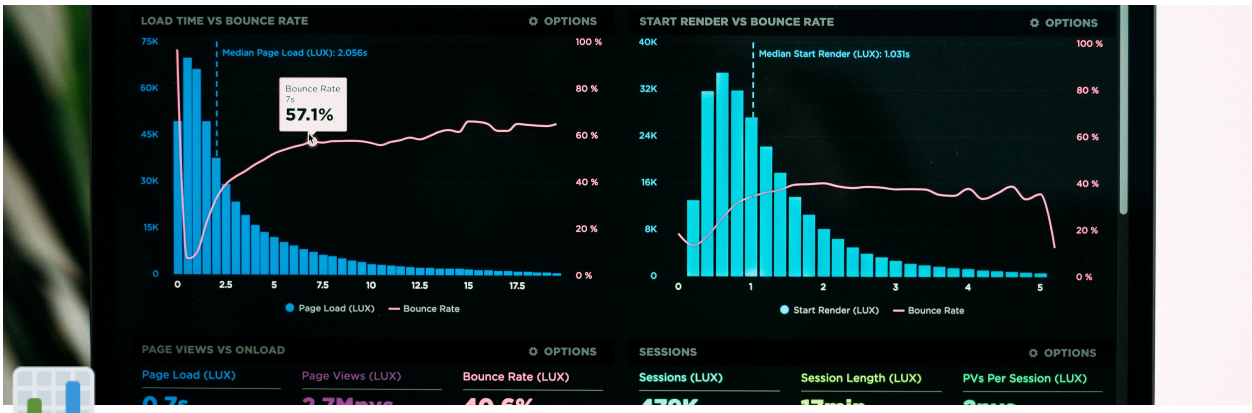


Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter, since then it has an active development community, and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell.

Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6.Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement.

Comaprating with MATLAB - MATPLOTLB
Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

Matplotlib is an excellent 2D and 3D graphics library for generating scientific figures. Some of the many advantages of this library include:

- Easy to get started
- Support for formatted labels and texts

  LATEX

  LATEX
- Great control of every element in a figure, including figure size and DPI.
- High-quality output in many formats, including PNG, PDF, SVG, EPS, and PGF.
- GUI for interactively exploring figures *and* support for headless generation of figure files (useful for batch jobs).

One of the key features of matplotlib that I would like to emphasize, and that I think makes matplotlib highly suitable for generating figures for scientific publications is that all aspects of the figure can be controlled *programmatically*. This is important for reproducibility and convenient when one needs to regenerate the figure with updated data or change its appearance.

Matplotlib Advantages

There are several advantages of using matplotlib to visualize data.

- A multi-platform data visualization tool built on the numpy and sidepy framework. Therefore, it's fast and efficient.
- It possesses the ability to work well with many operating systems and graphic backends.
- It possesses high-quality graphics and plots to print and view for a range of graphs such as histograms, bar charts, pie charts, scatter plots and heat maps.
- With Jupyter notebook integration, the developers have been free to spend their time implementing features rather than struggling with compatibility.
- It has large community support and cross-platform support as it is an open source tool.
- It has full control over graph or plot styles such as line properties, thoughts, and access properties.
- **It uses Python:** Python is a very interesting language for scientific purposes (it's interpreted, high-level, easy to learn, easily extensible, and has a powerful standard library) and is now used by major institutions such as NASA, JPL, Google, DreamWorks, Disney, and many more.
- **It's open source, so no license to pay:** This makes it very appealing for professors and students, who often have a low budget.
- **It's a real programming language:** The MATLAB language (while being Turing-complete) lacks many of the features of a general-purpose language like Python.
- **It's much more complete:** Python has a lot of external modules that will help us perform all the functions we need to. So it's the perfect tool to acquire data, elaborate the data, and then plot the data.
- **It's very customizable and extensible:** Matplotlib can fit every use case because it has a lot of graph types, features, and configuration options.
- **It's integrated with LaTeX markup:** This is really useful when writing scientific papers.
- **It's cross-platform and portable:** Matplotlib can run on Linux, Windows, Mac OS X, and Sun Solaris (and Python can run on almost every architecture available).

Useage

Some people use matplotlib interactively from the python shell and have plotting windows pop up when they type commands. Some people run Jupyter notebooks and draw inline plots for quick data analysis. Others embed matplotlib into graphical user interfaces like wxpython or pygtk to build rich applications.

Matplotlib Vs Seaborn

Matplotlib: Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots and so on. Seaborn: Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes.

References :-

https://analyticsindiamag.com/comparing-python-data-visualization-tools-matplotlib-vs-seaborn/

https://www.simplilearn.com/data-visualization-in-python-using-matplotlib-tutorial#:~:text=matplotlib Advantages&text=Therefore%2C it's fast and efficient,scatter plots and heat maps.

https://www.southampton.ac.uk/~feeg1001/notebooks/Matplotlib.html

https://www.activestate.com/blog/plotting-data-in-python-matplotlib-vs-plotly/

https://matplotlib.org/gallery/index.html

https://subscription.packtpub.com/book/application_development/9781847197900/1/ch01lvl1sec01/merits-of-matplotlib

Yet to Refer :-

https://www.kaggle.com/fazilbtopal/visualization-matplotlib-vs-seaborn

https://www.quora.com/What-is-the-difference-between-Matplotlib-and-Seaborn-Which-one-should-I-learn-for-studying-data-science

https://www.quora.com/Between-Matplotlib-vs-Seaborn-which-one-is-used-in-production-level-code

https://www.quora.com/Which-data-visualization-tool-should-I-learn-Matplotlib-Seaborn-Bokeh

https://www.quora.com/Which-is-better-Tableau-or-Seaborn-Matplotlib

https://www.quora.com/Is-Seaborn-a-better-visualisation-tool-than-matplotlib-and-why

https://www.quora.com/unanswered/I-have-self-studied-NumPy-Matplotlib-and-Pandas-What-library-of-Python-should-I-learn-next-for-data-science

https://blog.magrathealabs.com/choosing-one-of-many-python-visualization-tools-7eb36fa5855f

https://www.geeksforgeeks.org/difference-between-matplotlib-vs-seaborn/

Info Table

Sr.	Types of Graphs	Different Possible Graphs
1	Lines, bars and markers	44
2	Images, contours and fields	43
3	Subplots, axes and figures	32
4	Statistics	20
5	Pie and polar charts	9
6	Text, labels and annotations	40
7	Animation	12
8	3D plotting	36
9	Mnay more types	50+

CS Computer Science and Information Technology

Section 1: Engineering Mathematics

Discrete Mathematics: Propositional and first order logic. Sets, relations, functions, partial orders and lattices. Monoids, Groups. Graphs: connectivity, matching, coloring. Combinatorics: counting, recurrence relations, generating functions.

Linear Algebra: Matrices, determinants, system of linear equations, eigenvalues and eigenvectors, LU decomposition.

Calculus: Limits, continuity and differentiability. Maxima and minima. Mean value theorem. Integration.

Probability and Statistics: Random variables. Uniform, normal, exponential, poisson and binomial distributions. Mean, median, mode and standard deviation. Conditional probability and Bayes theorem.

Computer Science and Information Technology

Section 2: Digital Logic

Boolean algebra. Combinational and sequential circuits. Minimization. Number representations and computer arithmetic (fixed and floating point).

Section 3: Computer Organization and Architecture

Machine instructions and addressing modes. ALU, data-path and control unit. Instruction pipelining, pipeline hazards. Memory hierarchy: cache, main memory and secondary storage; I/O interface (interrupt and DMA mode).

Section 4: Programming and Data Structures

Programming in C. Recursion. Arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.

Section 5: Algorithms

Searching, sorting, hashing. Asymptotic worst case time and space complexity. Algorithm design techniques: greedy, dynamic programming and divide-and-conquer. Graph traversals, minimum spanning trees, shortest paths

Section 6: Theory of Computation

Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and contex-free languages, pumping lemma. Turing machines and undecidability.

Section 7: Compiler Design

Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimisation, Data flow analyses: constant propagation, liveness analysis, common subexpression elimination.

Section 8: Operating System

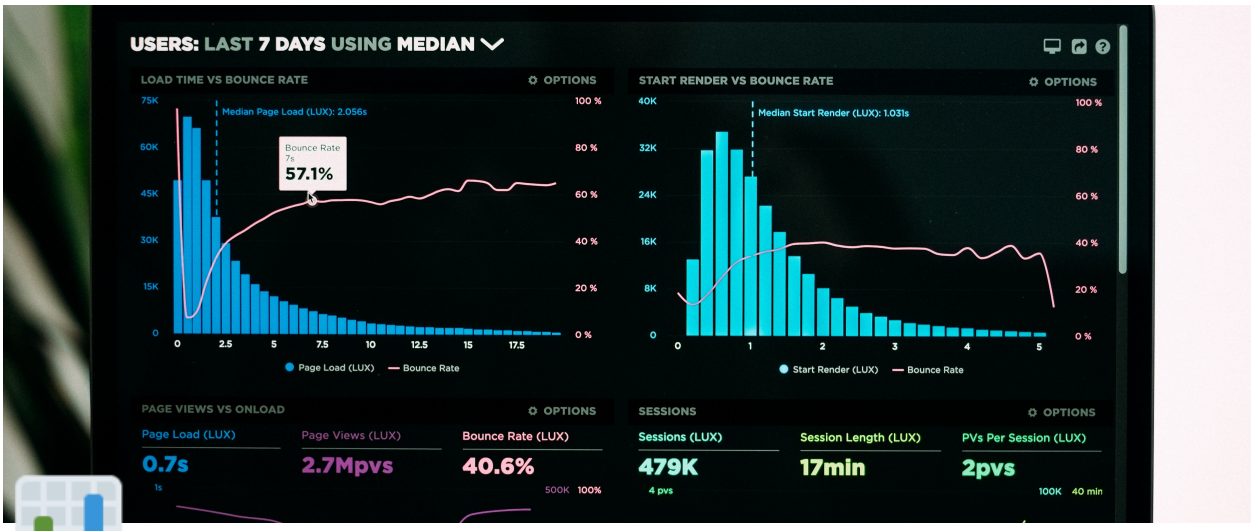
System calls, processes, threads, inter-process communication, concurrency and synchronization. Deadlock. CPU and I/O scheduling. Memory management and virtual memory. File systems.

Section 9: Databases

ER-model. Relational model: relational algebra, tuple calculus, SQL. Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control.

Section 10: Computer Networks

Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit-switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email.



Plotly (4.9.0 latest version)

The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.

Built on top of the Plotly JavaScript library (plotly.js), plotly enables Python users to create beautiful interactive web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash.

The plotly Python library is sometimes referred to as "plotly.py" to differentiate it from the
JavaScript library. Thanks to deep integration with the orca image export utility, plotly also provides great support for non-web contexts including desktop editors (e.g. QtConsole, Spyder, PyCharm) and static document publishing (e.g. exporting notebooks to PDF with high-quality vector images).

Type of graphs

Scatter plots,line charts,bar charts,pie charts,bubble charts,dot charts,filled area plots,horizontal bar charts,gantt charts,sunburst charts,tables,sankey diagram,treemap charts,webgl vs svg.

Advantages:

- It lets you create interactive visualizations built using D3.js without even having to
  know D3.js.
- It provides compatibility with number of different languages/ tools like R, Python,
  MATLAB, Perl, Julia, Arduino.
- Using plotly, interactive plots can easily be shared online with multiple people. Plotly can also be used by people with no technical background for creating interactive
  plots by uploading the data and using plotly GUI.
- Plotly is compatible with ggplots in R and Python.
- It allows to embed interactive plots in projects or websites using iframes or html.
- The syntax for creating interactive plots using plotly is very simple as well.

Disadvantages:

- The plots made using plotly community version are always public and can be viewed
  by anyone.
- For plotly community version, there is an upper limit on the API calls per day.

- There are also limited number of color Palettes available in community version
  which acts as an upper bound on the coloring options.

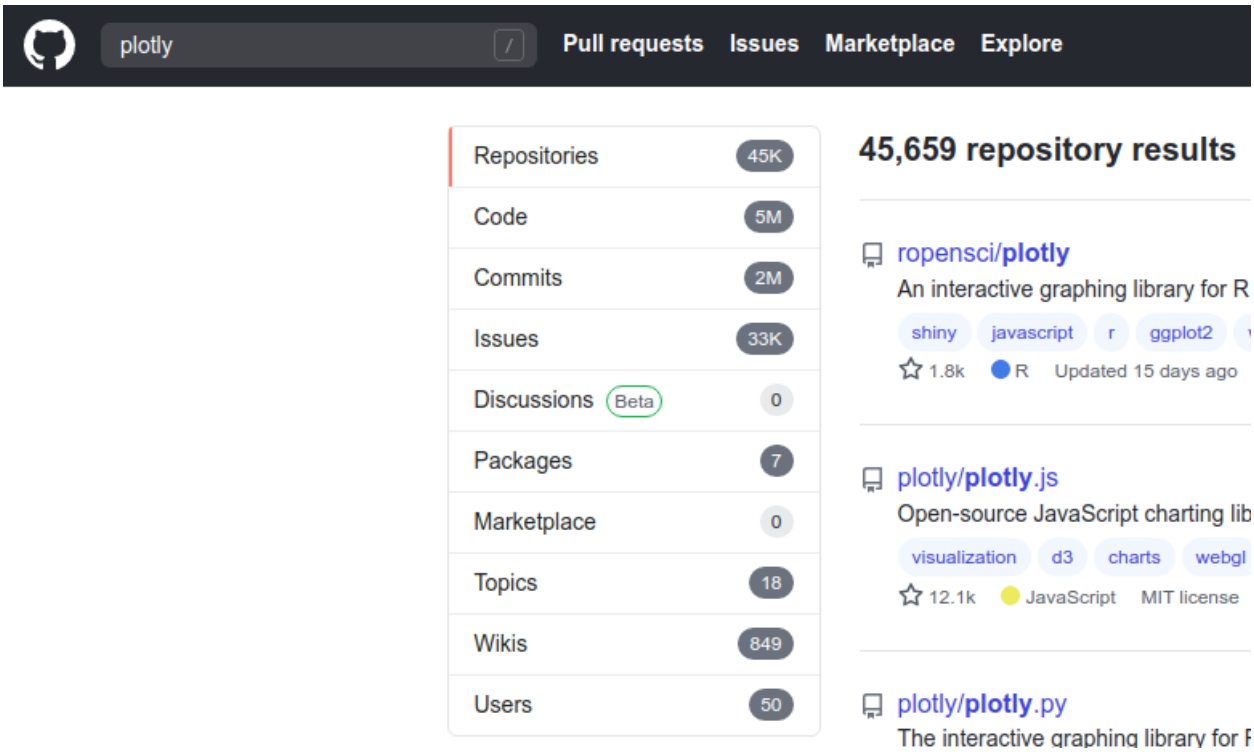
Basic Visualization:

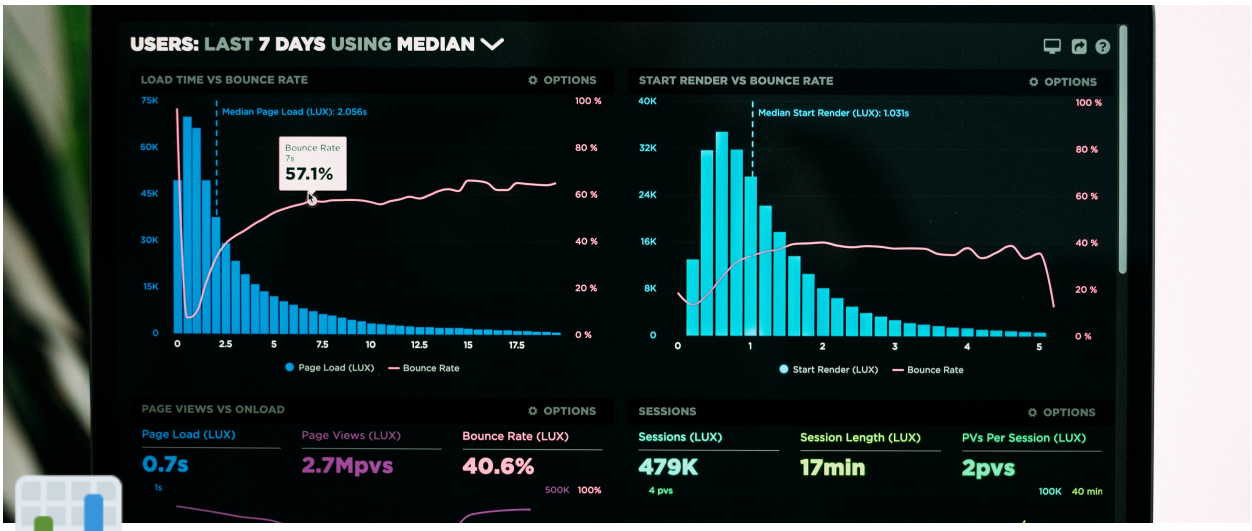
To get a good understanding of when you should use which plot, I'll recommend you to check
out this resource. Feel free to play around and explore these plots more. Here are a few things
that you can try in the coming plots:

- hovering your mouse over the plot to view associated attributes
- selecting a particular region on the plot using your mouse to zoom
- resetting the axis
- rotating the 3D images

▼ Basic Charts

Scatter Plot
Line Charts
Bar Chart
Pie Chart
Bubble Chart
Dot Plot
Filled Area Plots
Horizontal Bar Charts
Gantt Charts
Sunbrust Chart
Tables
Sankey Diagram
Treemap charts


plotly
/
Pull requests
Issues
Marketplace
Explore
Repositories 45K
Code 5M
Commits 2M
Issues 33K
Discussions Beta 0
Packages 7
Marketplace 0
Topics 18
Wikis 849
Users 50
45,659 repository results
ropensci/plotly
An interactive graphing library for R
shiny
javascript
r
ggplot2
1.8k
R
Updated 15 days ago
plotly/plotly.js
Open-source JavaScript charting lib
visualization
d3
charts
webgl
12.1k
JavaScript
MIT license
plotly/plotly.py
The interactive graphing library for



Seaborn (0.10.1 latest version)

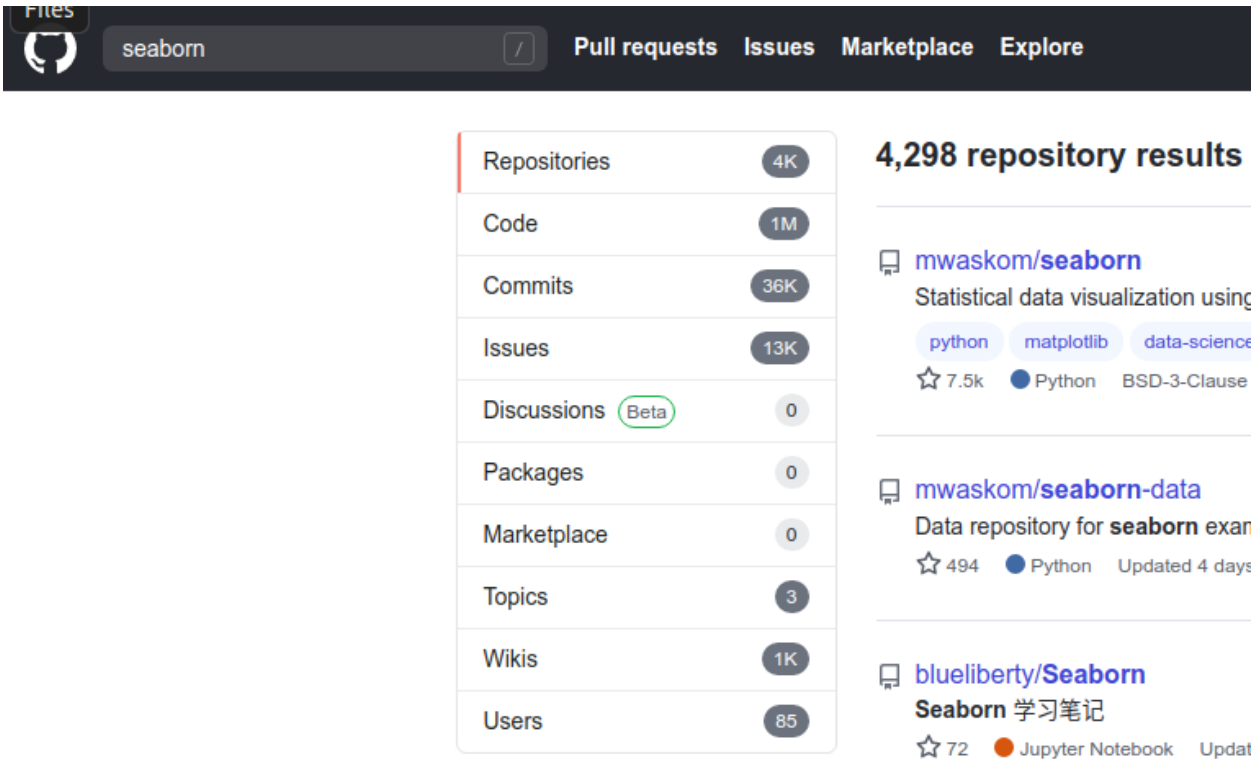
Seaborn is a python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics. It was originally developed at Stanford University and is widely used for plotting and visualizing data. There are several advantages:

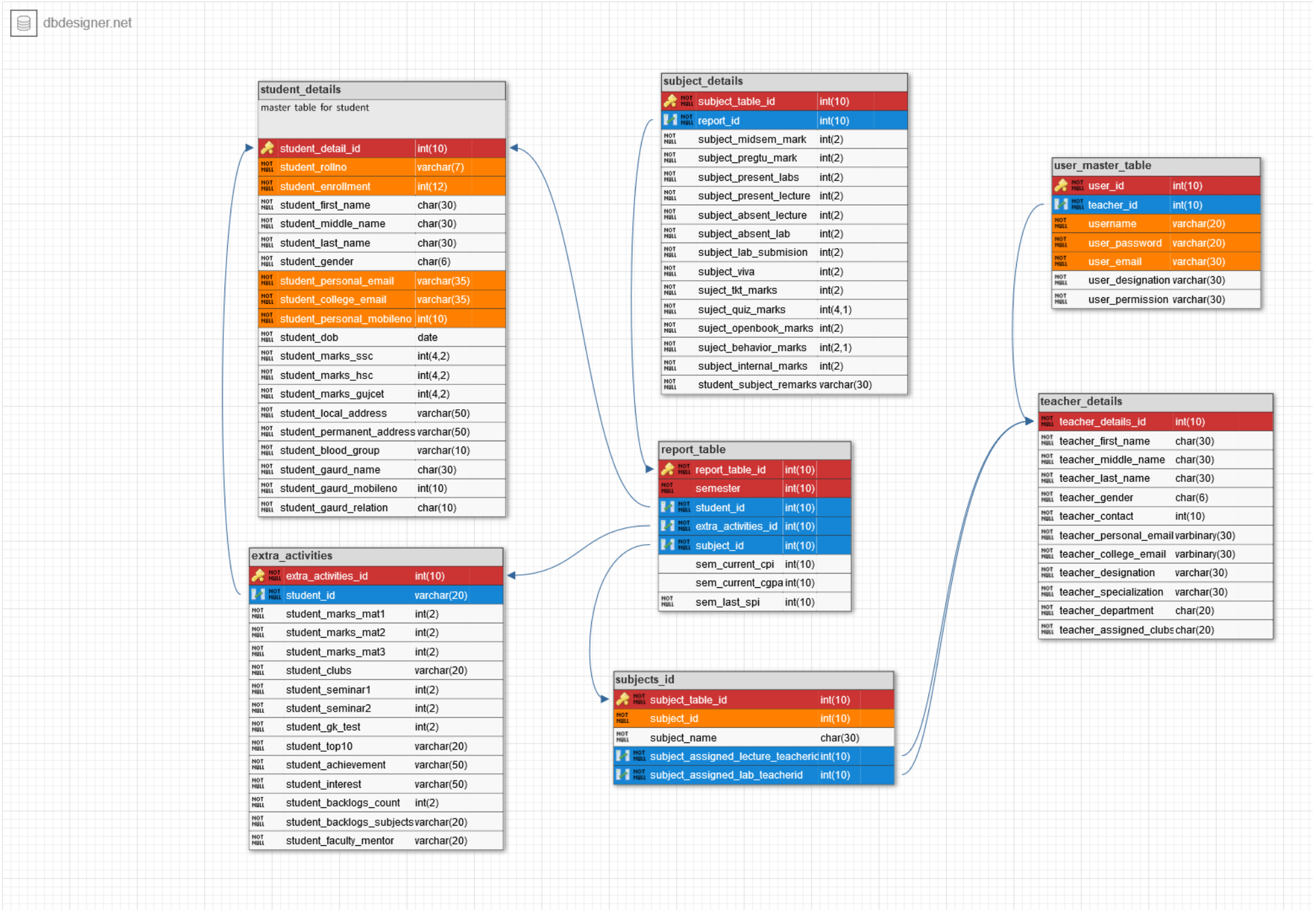
- It possesses built-in themes for better visualizations.
- It has tools built in statistical functions which reveal hidden patterns in the data set.
- It has functions to visualize matrices of data which become very important when visualizing large data sets.
- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data
- Automatic estimation and plotting of linear regression models for different kinds dependent variables
- Convenient views onto the overall structure of complex datasets

- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

Almost every graphs which can be generate in matplotlib also can be createed in seaborn with beutification in-built

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.




dbdesigner.net
student_details
master table for student
student_detail_id int(10)
student_rollno varchar(7)
student_enrollment int(12)
student_first_name char(30)
student_middle_name char(30)
student_last_name char(30)
student_gender char(6)
student_personal_email varchar(35)
student_college_email varchar(35)
student_personal_mobileno int(10)
student_dob date
student_marks_ssc int(4,2)
student_marks_hsc int(4,2)
student_marks_gujcet int(4,2)
student_local_address varchar(50)
student_permanent_address varchar(50)
student_blood_group varchar(10)
student_gaurd_name char(30)
student_gaurd_mobileno int(10)
student_gaurd_relation char(10)
subject_details
subject_table_id int(10)
report_id int(10)
subject_midsem_mark int(2)
subject_pregtu_mark int(2)
subject_present_labs int(2)
subject_present_lecture int(2)
subject_absent_lecture int(2)
subject_absent_lab int(2)
subject_lab_submision int(2)
subject_viva int(2)
suject_tkt_marks int(2)
suject_quiz_marks int(4,1)
suject_openbook_marks int(2)
suject_behavior_marks int(2,1)
subject_internal_marks int(2)
student_subject_remarks varchar(30)
user_master_table
user_id int(10)
teacher_id int(10)
username varchar(20)
user_password varchar(20)
user_email varchar(30)
user_designation varchar(30)
user_permission varchar(30)
report_table
report_table_id int(10)
semester int(10)
student_id int(10)
extra_activities_id int(10)
subject_id int(10)
sem_current_cpi int(10)
sem_current_cgpa int(10)
sem_last_spi int(10)
teacher_details
teacher_details_id int(10)
teacher_first_name char(30)
teacher_middle_name char(30)
teacher_last_name char(30)
teacher_gender char(6)
teacher_contact int(10)
teacher_personal_email varbinary(30)
teacher_college_email varbinary(30)
teacher_designation varchar(30)
teacher_specialization varchar(30)
teacher_department char(20)
teacher_assigned_clubs char(20)
extra_activities
extra_activities_id int(10)
student_id varchar(20)
student_marks_mat1 int(2)
student_marks_mat2 int(2)
student_marks_mat3 int(2)
student_clubs varchar(20)
student_seminar1 int(2)
student_seminar2 int(2)
student_gk_test int(2)
student_top10 varchar(20)
student_achievement varchar(50)
student_interest varchar(50)
student_backlogs_count int(2)
student_backlogs_subjects varchar(20)
student_faculty_mentor varchar(20)
subjects_id
subject_table_id int(10)
subject_id int(10)
subject_name char(30)
subject_assigned_lecture_teacherid int(10)
subject_assigned_lab_teacherid int(10)

HR Questions

1. Tell me sometabout yourself in brief.

2. What are your strengths and weaknesses? With examples

3. Explain the difference between group and team. Are you a team player?

4. What is the difference between hard work and smart work?

5. How do you feel about working weekends and night shifts?

6. Where do you see yourself in 5 years?

7. On a scale of 1 to 10 how would you rate yourself as a leader?

8. So can you give any example of it that you have led some people or else?

9. What are your future goals?Tell me about your short & long-term goals.

10. What are the three things that are most important for you in a job?