

python gui for data analysis

Python GUI for Data Analysis: Unlocking Interactive Insights with Visual Tools

python gui for data analysis has become an increasingly popular approach for data scientists, analysts, and developers who want to combine the power of Python's data processing capabilities with intuitive, user-friendly interfaces. While Python's command-line environment and scripting prowess are well-known, creating graphical user interfaces (GUIs) opens up new possibilities for exploring data, presenting results, and building interactive dashboards that anyone can use—regardless of their coding background.

In this article, we'll dive into the world of Python GUIs tailored specifically for data analysis, exploring why they matter, which tools are best suited for the job, and practical tips for building efficient and engaging applications. Whether you're a seasoned programmer or just starting out, understanding how to integrate GUI elements with Python's rich data ecosystem can significantly enhance your workflow and the impact of your data insights.

Why Use a Python GUI for Data Analysis?

When working with data, it's common to write scripts that crunch numbers, generate plots, or export reports. However, these scripts often require users to interact through the command line or edit code directly, which can be limiting for collaboration or presentation. A Python GUI bridges this gap by providing an interactive interface where users can:

- Select datasets dynamically
- Adjust parameters via sliders, dropdowns, or input fields
- Visualize data through embedded charts and graphs
- Export results easily without diving into code

This convenience not only improves user experience but also makes data analysis more accessible to non-technical stakeholders. Imagine a business analyst who can tweak filters on a sales dashboard without needing to understand pandas or matplotlib commands—that's the power of a well-crafted GUI.

Popular Python GUI Frameworks for Data Analysis

Choosing the right GUI toolkit can make or break your data application's usability and performance. Here are some of the leading Python GUI frameworks that excel in building data analysis tools:

1. Tkinter

Tkinter is Python's standard GUI library, included with most Python installations. It is lightweight and straightforward, making it a great starting point for beginners. While Tkinter's widgets are somewhat basic compared to modern frameworks, it supports essential components like buttons, labels, sliders, and canvases.

For data analysis, Tkinter can be combined with plotting libraries such as Matplotlib to embed interactive charts within the GUI. Its simplicity means faster development cycles, but for highly polished or complex interfaces, you might find it limiting.

2. PyQt and PySide

PyQt and PySide are bindings for the Qt framework, offering a rich set of widgets, advanced styling options, and cross-platform compatibility. These frameworks are ideal for creating professional-grade applications with sophisticated layouts and interactive elements.

In data analysis projects, PyQt/PySide shine because they integrate well with libraries like pandas, NumPy, and Matplotlib. Features such as table views and model-view architecture make it easier to display large datasets interactively. The learning curve is steeper than Tkinter, but the payoff is a highly customizable and scalable GUI.

3. Kivy

Kivy is an open-source Python library designed for developing multitouch applications and is particularly useful if you want your data analysis tools to run on mobile devices or touchscreens. Its modern UI elements and flexibility in design allow for creative interfaces that go beyond traditional desktop paradigms.

While it may require more effort to connect Kivy apps with standard data visualization tools, its ability to create dynamic and responsive interfaces makes it a unique choice for certain interactive data projects.

4. Dash by Plotly

Dash is a specialized framework built on top of Flask, React.js, and Plotly.js that focuses specifically on creating analytical web applications. Unlike traditional desktop GUIs, Dash apps run in a web browser, which means they are inherently cross-platform and easy to share.

Dash is particularly popular for data analysis because it combines Python's data science libraries with interactive plots, sliders, dropdowns, and live updates—all without requiring deep front-end development knowledge. It's an excellent tool for building dashboards or reports that update in real-time.

Integrating Data Visualization in Python GUIs

Visualization is a cornerstone of data analysis, and embedding dynamic plots into GUI applications greatly enhances user experience. Here's how some popular plotting libraries pair with Python GUIs:

Matplotlib

Matplotlib is the most widely used plotting library in Python. It can be embedded into GUIs using backends like `FigureCanvasTkAgg` for Tkinter or `FigureCanvasQTAgg` for PyQt. This allows you to create static or interactive charts within your application window.

Tips for better integration:

- Use Matplotlib's interactive navigation toolbar for zooming and panning
- Combine with widgets to update plots based on user input dynamically
- Keep plot redraws efficient by updating existing figures instead of recreating them

Plotly

Plotly is a powerful library for creating interactive, web-based plots that support features like hovering, zooming, and real-time updates. Dash leverages Plotly directly, but you can also embed Plotly graphs in PyQt or Tkinter GUIs using web views (e.g., `QWebEngineView`).

Plotly's interactive nature makes it ideal for exploratory data analysis

where users want to drill down into the data visually without losing detail.

Seaborn and Bokeh

Seaborn builds on Matplotlib to provide high-level statistical graphics with attractive default styles. While it's primarily for static plots, you can embed Seaborn charts in GUIs much like Matplotlib.

Bokeh, on the other hand, specializes in interactive visualizations and can be integrated into web-based GUIs or embedded in desktop apps using HTML rendering components.

Best Practices for Building Python GUIs for Data Analysis

Creating an effective GUI for data analysis isn't just about slapping buttons and charts together. Thoughtful design and coding practices can make your application more robust and enjoyable to use.

Focus on User Experience

- Keep interfaces clean and uncluttered; avoid overwhelming users with too many controls.
- Use descriptive labels and tooltips to guide users through data selection and analysis options.
- Provide immediate feedback when actions are taken, such as loading indicators when processing large datasets.

Ensure Responsiveness

Data operations can be resource-intensive. To prevent your GUI from freezing:

- Use multithreading or multiprocessing to run data computations separately from the main UI thread.
- Display progress bars or status messages during long-running tasks.
- Optimize data processing pipelines to minimize delays.

Modularize Your Code

Separate GUI logic from data analysis functions. This organization helps in:

- Maintaining and debugging your application
- Reusing analysis code in other projects or scripts
- Improving collaboration when frontend and backend development are handled by different people

Real-World Applications of Python GUIs in Data Analysis

Python GUI applications for data analysis aren't just academic exercises; they power many real-world solutions across industries.

Financial Data Dashboards

Investment firms and traders use Python GUIs to monitor stock prices, run simulations, and visualize market trends in real-time. Dashboards built with Dash or PyQt can integrate live data feeds, allowing quick decision-making.

Scientific Research Tools

Researchers often create custom GUIs to process experimental data, visualize results, and export summaries. For example, Tkinter combined with Matplotlib can help biologists analyze microscopy images or physicists plot sensor readings.

Business Intelligence Applications

Non-technical managers can interact with sales data, customer feedback, or operational reports through Python-based GUIs that simplify complex datasets into actionable insights. Frameworks like Dash facilitate sharing interactive reports via web browsers.

Getting Started: A Simple Python GUI Example for Data Analysis

To illustrate how these pieces come together, here's a quick conceptual example using Tkinter and Matplotlib:

```
```python
import tkinter as tk
from tkinter import filedialog
import pandas as pd
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt

def load_data():
 file_path = filedialog.askopenfilename()
 if file_path:
 df = pd.read_csv(file_path)
 plot_data(df)

def plot_data(df):
 ax.clear()
 df.plot(ax=ax)
 canvas.draw()

root = tk.Tk()
root.title("Simple Data Analysis GUI")

load_button = tk.Button(root, text="Load CSV Data", command=load_data)
load_button.pack()

fig, ax = plt.subplots(figsize=(5,4))
canvas = FigureCanvasTkAgg(fig, master=root)
canvas.get_tk_widget().pack()

root.mainloop()
```
```

This basic application lets users load a CSV file and immediately see a plot of the data. From here, you could expand the interface with dropdowns to select columns, sliders to filter data ranges, or buttons to perform statistical summaries.

Exploring Python GUI frameworks alongside data analysis libraries can transform how you interact with data—making analysis more accessible, interactive, and visually compelling. Whether building simple tools or full-fledged dashboards, the combination of Python GUIs and data science unlocks new dimensions for data-driven storytelling and decision-making.

Frequently Asked Questions

What are the best Python GUI frameworks for data analysis applications?

Some of the best Python GUI frameworks for data analysis include PyQt, Tkinter, Kivy, and wxPython. PyQt is popular for its rich features and native look, while Tkinter is simple and comes bundled with Python. Kivy is suitable for multi-touch applications, and wxPython offers native UI components.

How can I integrate matplotlib plots into a Python GUI for data analysis?

You can integrate matplotlib plots into Python GUI frameworks like PyQt or Tkinter by embedding the matplotlib FigureCanvas widget into the GUI layout. This allows interactive plotting within the application, enabling dynamic visualization of data.

Is it possible to create interactive dashboards for data analysis using Python GUI libraries?

Yes, you can create interactive dashboards using Python GUI libraries such as Dash (built on Flask and React), PyQt, or Tkinter combined with plotting libraries. Dash is especially designed for building web-based interactive dashboards, while PyQt and Tkinter can be used for desktop applications.

What are the advantages of using PyQt over Tkinter for data analysis GUIs?

PyQt offers a more modern and feature-rich set of widgets, better cross-platform consistency, and support for advanced UI design compared to Tkinter. It also supports more complex layouts and customizations, making it suitable for sophisticated data analysis applications.

How can I handle large datasets efficiently in a Python GUI application for data analysis?

To handle large datasets efficiently, use data processing libraries like pandas and NumPy to preprocess data before visualization. Employ techniques like data pagination, lazy loading, or background threading to prevent the GUI from freezing and maintain responsiveness.

Can Python GUI applications for data analysis be deployed as standalone executables?

Yes, Python GUI applications can be packaged as standalone executables using

tools like PyInstaller, cx_Freeze, or py2exe. This allows users to run the application without needing to install Python or dependencies separately.

What role does Jupyter Notebook play in Python GUI for data analysis?

Jupyter Notebook provides an interactive environment for data analysis, but it is not a traditional GUI framework. However, you can use widgets like ipywidgets to create interactive controls within notebooks, which can serve as lightweight GUIs for data exploration.

Are there any Python GUI tools specifically designed for real-time data analysis?

Yes, frameworks like PyQt and Kivy support real-time data visualization and interaction. Additionally, libraries such as Bokeh and Dash enable real-time updates in web-based dashboards, which can be integrated with Python GUI applications for dynamic data analysis.

How do I choose the right Python GUI framework for my data analysis project?

Consider factors such as project complexity, target platform, ease of use, community support, and the type of data visualization needed. For simple tools, Tkinter may suffice; for complex, feature-rich applications, PyQt or Dash might be better. Evaluate your requirements and experiment with frameworks before deciding.

Additional Resources

Python GUI for Data Analysis: Exploring Tools and Techniques for Enhanced Interactivity

python gui for data analysis has become an increasingly relevant topic in the data science community, especially as professionals seek more interactive and user-friendly ways to manipulate and visualize data. While Python is renowned for its powerful libraries like Pandas, NumPy, and Matplotlib, the addition of graphical user interfaces (GUIs) transforms data analysis workflows by making them accessible to users who may not be comfortable with command-line inputs or scripting. This article delves into the landscape of Python GUI frameworks tailored for data analysis, examining their capabilities, use cases, and how they integrate with data processing libraries.

The Role of Python GUI in Modern Data Analysis

Data analysis traditionally relies heavily on scripts and notebooks, where analysts write code to clean, transform, and visualize datasets. However, complex scripts can become cumbersome and difficult to share with stakeholders unfamiliar with programming. Python GUIs bridge this gap by offering interactive environments where users can input parameters, trigger computations, and view results dynamically.

The integration of Python GUI frameworks with data analysis libraries allows for the creation of bespoke applications that can streamline workflows, reduce errors, and accelerate decision-making. These graphical front-ends often encapsulate complex logic behind simple controls like buttons, sliders, and dropdown menus.

Popular Python GUI Frameworks for Data Analysis

Several Python libraries stand out when it comes to developing GUIs specifically for data analysis purposes. Their varying degrees of complexity, flexibility, and learning curves make them suitable for different project requirements.

- **Tkinter:** As Python's standard GUI toolkit, Tkinter is lightweight and easy to learn. It can be used to create simple interfaces for data input and visualization. However, its basic widgets and limited styling options may not suffice for more sophisticated applications.
- **PyQt / PySide:** These are bindings to the Qt application framework, offering a rich set of widgets and advanced features. PyQt and PySide support complex layouts, custom widgets, and integration with plotting libraries like Matplotlib or Plotly, making them ideal for comprehensive data analysis tools.
- **Kivy:** Known for its ability to build multi-touch applications, Kivy is cross-platform and supports rapid prototyping. While it's less conventional for desktop data analysis, its flexibility can be advantageous for creating innovative interactive dashboards.
- **Dash:** Developed by Plotly, Dash is a modern framework that turns Python scripts into web-based analytical applications. It combines Flask for the backend, React for the frontend, and integrates seamlessly with Plotly's graphing libraries, facilitating highly interactive and visually appealing data analysis GUIs.
- **Streamlit:** Streamlit is designed to create data apps quickly without requiring frontend development expertise. It allows analysts to build interactive interfaces with minimal code, focusing on data visualization and exploration tasks.

Integration of Data Analysis Libraries with GUI Frameworks

Python's data analysis ecosystem is rich and mature, with libraries that handle everything from data wrangling to statistical modeling. Effective Python GUI applications for data analysis leverage these libraries to perform backend computations and generate visualizations.

For instance, Pandas DataFrames can be manipulated through GUI controls, enabling users to filter or aggregate data dynamically. Visualization libraries like Matplotlib, Seaborn, or Plotly can then render charts that update in real-time as inputs change.

In frameworks like PyQt, developers often embed Matplotlib plots directly into the interface using canvas widgets. Dash and Streamlit take a different approach by serving the interface through a browser, where interactive graphs respond to user inputs via callbacks. This seamless integration enhances the responsiveness and usability of data analysis tools.

Comparative Analysis: Choosing the Right Python GUI for Data Analysis

Selecting a GUI framework depends on various factors, including project complexity, target audience, deployment environment, and development time. Below is an analytical overview of key considerations:

Ease of Use vs. Flexibility

Tkinter offers the simplest entry point but lacks advanced features and appealing aesthetics, which can be a drawback for professional applications. PyQt/PySide provide extensive customization and robust widgets but require more time to master.

Dash and Streamlit strike a balance by abstracting frontend development complexities, letting users focus on Python code. They are particularly well-suited for data scientists who want to build interactive dashboards swiftly without deep GUI programming knowledge.

Deployment and Accessibility

Desktop-based GUIs like those built with Tkinter or PyQt are ideal for standalone applications requiring offline access. However, distributing these apps across different operating systems may involve packaging challenges.

Web-based frameworks such as Dash and Streamlit facilitate easy sharing through URLs and support collaborative environments. Their reliance on web browsers removes the need for end-users to install specialized software, which is a significant advantage in enterprise settings.

Performance and Scalability

For small to medium datasets and moderate user interaction, all frameworks perform adequately. When handling large datasets or complex visualizations, web-based frameworks might introduce latency due to network communication and rendering overhead.

PyQt applications, being native desktop programs, often deliver snappier performance, especially when optimized. However, they may require more resources and development effort to scale features.

Implementing a Basic Python GUI for Data Analysis: A Practical Overview

To illustrate the practical potential of Python GUI for data analysis, consider a simple example using PyQt5 combined with Pandas and Matplotlib.

The application could allow users to:

1. Load a CSV file through a file dialog.
2. Display the data in a table widget for inspection.
3. Select columns to plot via dropdown menus.
4. Render an interactive plot embedded within the GUI.

Such an interface provides a clear advantage over running standalone scripts, as it reduces the need for manual coding when performing routine analysis tasks. Adding features like filtering data rows or exporting plots enhances usability further.

Similarly, a Dash application could replicate this functionality on the web, with dropdown menus triggering callbacks to update charts dynamically without page refreshes. Streamlit's declarative syntax allows rapid construction of this workflow with minimal code, making it attractive for prototyping and sharing insights quickly.

Challenges in Developing Python GUIs for Data Analysis

While the benefits are clear, developing effective Python GUIs for data analysis is not without challenges:

- **Complexity of UI Design:** Balancing functionality with intuitive design requires careful planning, especially when targeting non-technical users.
- **Performance Bottlenecks:** Handling large datasets interactively can strain resources and necessitate optimization techniques like lazy loading or server-side processing.
- **Cross-platform Compatibility:** Ensuring consistent behavior across Windows, macOS, and Linux may require additional testing and adjustments.
- **Maintenance and Scalability:** As data analysis requirements evolve, maintaining and extending GUI applications can become resource-intensive.

Addressing these challenges involves leveraging community best practices, choosing appropriate frameworks, and sometimes combining multiple technologies to achieve the desired user experience.

The Future Landscape of Python GUI in Data Analysis

Advancements in Python GUI development continue to align with trends in data science, such as increased interactivity, real-time analytics, and collaborative workflows. Emerging tools are focusing on lowering the barrier to entry for creating rich data applications, often by integrating machine learning models and cloud services.

Moreover, hybrid approaches that blend desktop and web technologies are gaining traction, allowing developers to harness the strengths of both domains. For example, Electron-based applications that embed web GUIs offer native-like performance with the flexibility of browser-based interfaces.

As organizations demand more accessible and customizable data analysis solutions, the role of Python GUI frameworks will likely expand, empowering analysts and decision-makers alike to interact with data more effectively.

In summary, exploring Python GUI for data analysis reveals a vibrant ecosystem where developers can tailor applications to diverse needs—from simple visualization tools to complex analytical platforms. The choice of framework hinges on balancing development effort, user requirements, and deployment considerations, but the growing maturity of these tools ensures that interactive data analysis is becoming more accessible than ever before.

[Python Gui For Data Analysis](#)

python gui for data analysis: FIVE PROJECTS: POSTGRESQL AND PYTHON GUI FOR DATA ANALYSIS Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-11-03 PROJECT 1: ZERO TO MASTERY: THE COMPLETE GUIDE TO LEARNING POSTGRESQL WITH PYTHON GUI This book uses the PostgreSQL version of MySQL-based Northwind database. The Northwind database is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called “Northwind Traders,” which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind database has since been ported to a variety of non-Microsoft databases, including PostgreSQL. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; and Orders and Order_Details: Sales Order transactions taking place between the customers & the company. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature. PROJECT 2: FULL SOURCE CODE: POSTGRESQL AND DATA SCIENCE FOR PROGRAMMERS WITH PYTHON GUI This project uses the PostgreSQL version of MySQL-based Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the database from <https://dev.mysql.com/doc/sakila/en/>. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which store have most sales; plot average payment

amount by month with mean and EWM; and plot payment amount over June 2005. PROJECT 3: FULL SOURCE CODE: POSTGRESQL FOR DATA ANALYTICS AND VISUALIZATION WITH PYTHON GUI In this project, we provide you with a PostgreSQL version of an Oracle sample database named OT which is based on a global fictitious company that sells computer hardware including storage, motherboard, RAM, video card, and CPU. The company maintains the product information such as name, description standard cost, list price, and product line. It also tracks the inventory information for all products including warehouses where products are available. Because the company operates globally, it has warehouses in various locations around the world. The company records all customer information including name, address, and website. Each customer has at least one contact person with detailed information including name, email, and phone. The company also places a credit limit on each customer to limit the amount that customer can owe. Whenever a customer issues a purchase order, a sales order is created in the database with the pending status. When the company ships the order, the order status becomes shipped. In case the customer cancels an order, the order status becomes canceled. In addition to the sales information, the employee data is recorded with some basic information such as name, email, phone, job title, manager, and hire date. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by status, top 10 sales by status, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2016, amount feature over 2017, and amount payment in all years.

PROJECT 4: FULL SOURCE CODE: POSTGRESQL FOR DATA SCIENTISTS AND DATA ANALYSTS WITH PYTHON GUI In this project, we will use the PostgreSQL version of SQL Server based BikeStores as a sample database to help you work with PostgreSQL quickly and effectively. The detailed structure of database can be found at:

<https://www.sqlservertutorial.net/sql-server-sample-database/>. The stores table includes the store's information. Each store has a store name, contact information such as phone and email, and an address including street, city, state, and zip code. The staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone. A staff works at a store specified by the value in the store_id column. A store can have one or more staffs. A staff reports to a store manager specified by the value in the manager_id column. If the value in the manager_id is null, then the staff is the top manager. If a staff no longer works for any stores, the value in the active column is set to zero. The categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes. The products table stores the product's information such as name, brand, category, model year, and list price. Each product belongs to a brand specified by the brand_id column. Hence, a brand may have zero or many products. Each product also belongs a category specified by the category_id column. Also, each category may have zero or many products. The customers table stores customer's information including first name, last name, phone, email, street, city, state, zip code, and photo path. The orders table stores the sales order's header information including customer, order status, order date, required date, shipped date. It also stores the information on where the sales transaction was created (store) and who created it (staff). Each sales order has a row in the sales_orders table. A sales order has one or many line items stored in the order_items table. The order_items table stores the line items of a sales order. Each line item belongs to a sales order specified by the order_id column. A sales order line item includes product, order quantity, list price, and discount. The stocks table stores the inventory information i.e. the quantity of a particular product in a specific store. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to

plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by brand, top 10 sales by brand, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2017, amount feature over 2018, and all amount feature. PROJECT 5: FULL SOURCE CODE: THE COMPLETE GUIDE TO LEARNING POSTGRESQL AND DATA SCIENCE WITH PYTHON GUI In this project, we provide you with the PostgreSQL version of SQLite sample database named chinook. The chinook sample database is a good database for practicing with SQL, especially PostgreSQL. The detailed description of the database can be found on: <https://www.sqlitetutorial.net/sqlite-sample-database/>. The sample database consists of 11 tables: The employee table stores employees data such as employee id, last name, first name, etc. It also has a field named ReportsTo to specify who reports to whom; customers table stores customers data; invoices & invoice_items tables: these two tables store invoice data. The invoice table stores invoice header data and the invoice_items table stores the invoice line items data; The artist table stores artists data. It is a simple table that contains only the artist id and name; The album table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums; The media_type table stores media types such as MPEG audio and AAC audio files; genre table stores music types such as rock, jazz, metal, etc; The track table stores the data of songs. Each track belongs to one album; playlist & playlist_track tables: The playlist table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the playlist table and track table is many-to-many. The playlist_track table is used to reflect this relationship. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the bottom/top 10 sales by employee, the bottom/top 10 sales by customer, the bottom/top 10 sales by customer, the bottom/top 10 sales by artist, the bottom/top 10 sales by genre, the bottom/top 10 sales by play list, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the payment amount by month with mean and EWM, the average payment amount by every month, and amount payment in all years.

python gui for data analysis: THREE PROJECTS: SQL SERVER AND PYTHON GUI FOR DATA ANALYSIS Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-11-08 PROJECT 1: FULL SOURCE CODE: SQL SERVER FOR STUDENTS AND DATA SCIENTISTS WITH PYTHON GUI In this project, we provide you with the SQL SERVER version of SQLite sample database named chinook. The chinook sample database is a good database for practicing with SQL, especially PostgreSQL. The detailed description of the database can be found on: <https://www.sqlitetutorial.net/sqlite-sample-database/>. The sample database consists of 11 tables: The employee table stores employees data such as employee id, last name, first name, etc. It also has a field named ReportsTo to specify who reports to whom; customers table stores customers data; invoices & invoice_items tables: these two tables store invoice data. The invoice table stores invoice header data and the invoice_items table stores the invoice line items data; The artist table stores artists data. It is a simple table that contains only the artist id and name; The album table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums; The media_type table stores media types such as MPEG audio and AAC audio files; genre table stores music types such as rock, jazz, metal, etc; The track table stores the data of songs. Each track belongs to one album; playlist & playlist_track tables: The playlist table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the playlist table and track table is many-to-many. The playlist_track table is used to reflect this relationship. In this project, you will write Python script to create every table and

insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the bottom/top 10 sales by employee, the bottom/top 10 sales by customer, the bottom/top 10 sales by customer, the bottom/top 10 sales by artist, the bottom/top 10 sales by genre, the bottom/top 10 sales by play list, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the payment amount by month with mean and EWM, the average payment amount by every month, and amount payment in all years. PROJECT 2: FULL SOURCE CODE: SQL SERVER FOR DATA ANALYTICS AND VISUALIZATION WITH PYTHON GUI This book uses SQL SERVER version of MySQL-based Sakila sample database. It is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, customer, rental, payment and inventory among others. The Sakila sample database is intended to provide a standard schema that can be used for examples in books, tutorials, articles, samples, and so forth. Detailed information about the database can be found on website:

<https://dev.mysql.com/doc/index-other.html>. In this project, you will develop GUI using PyQt5 to: read SQL SERVER database and every table in it; read every actor in actor table, read every film in films table; plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue customers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. PROJECT 3: ZERO TO MASTERY: THE COMPLETE GUIDE TO LEARNING SQL SERVER AND DATA SCIENCE WITH PYTHON GUI In this project, we provide you with a SQL SERVER version of an Oracle sample database named OT which is based on a global fictitious company that sells computer hardware including storage, motherboard, RAM, video card, and CPU. The company maintains the product information such as name, description standard cost, list price, and product line. It also tracks the inventory information for all products including warehouses where products are available. Because the company operates globally, it has warehouses in various locations around the world. The company records all customer information including name, address, and website. Each customer has at least one contact person with detailed information including name, email, and phone. The company also places a credit limit on each customer to limit the amount that customer can owe. Whenever a customer issues a purchase order, a sales order is created in the database with the pending status. When the company ships the order, the order status becomes shipped. In case the customer cancels an order, the order status becomes canceled. In addition to the sales information, the employee data is recorded with some basic information such as name, email, phone, job title, manager, and hire date. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by status, top 10 sales by status, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2016, amount feature over 2017, and amount payment in all years.

python gui for data analysis: FIVE PROJECTS: SQLITE AND PYTHON GUI FOR DATA

ANALYSIS Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-11-03 PROJECT 1: FULL SOURCE CODE: PRACTICAL DATA SCIENCE WITH SQLITE AND PYTHON GUI In this project, we provide you with the SQLite sample database named chinook. The chinook sample database is a good database for practicing with SQL, especially SQLite. The detailed description of the database can be found on: <https://www.sqlitetutorial.net/sqlite-sample-database/>. There are 11 tables in the chinook sample database: The employee table stores employees data such as employee id, last name, first name, etc. It also has a field named ReportsTo to specify who reports to whom; customers table stores customers data; invoices & invoice_items tables: these two tables store invoice data. The invoice table stores invoice header data and the invoice_items table stores the invoice line items data; The artist table stores artists data. It is a simple table that contains only the artist id and name; The album table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums; The media_type table stores media types such as MPEG audio and AAC audio files; genre table stores music types such as rock, jazz, metal, etc; The track table stores the data of songs. Each track belongs to one album; playlist & playlist_track tables: The playlist table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the playlist table and track table is many-to-many. The playlist_track table is used to reflect this relationship. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the bottom/top 10 sales by employee, the bottom/top 10 sales by customer, the bottom/top 10 sales by customer, the bottom/top 10 sales by artist, the bottom/top 10 sales by genre, the bottom/top 10 sales by play list, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the payment amount by month with mean and EWM, the average payment amount by every month, and amount payment in all years. PROJECT 2: FULL SOURCE CODE: SQLITE FOR STUDENTS AND PROGRAMMERS WITH PYTHON GUI In this project, we provide you with a SQLITE version of an Oracle sample database named OT which is based on a global fictitious company that sells computer hardware including storage, motherboard, RAM, video card, and CPU. You can find the detailed structures of the database: <https://www.oracletutorial.com/getting-started/oracle-sample-database/>. The company maintains the product information such as name, description standard cost, list price, and product line. It also tracks the inventory information for all products including warehouses where products are available. Because the company operates globally, it has warehouses in various locations around the world. The company records all customer information including name, address, and website. Each customer has at least one contact person with detailed information including name, email, and phone. The company also places a credit limit on each customer to limit the amount that customer can owe. Whenever a customer issues a purchase order, a sales order is created in the database with the pending status. When the company ships the order, the order status becomes shipped. In case the customer cancels an order, the order status becomes canceled. In addition to the sales information, the employee data is recorded with some basic information such as name, email, phone, job title, manager, and hire date. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by status, top 10 sales by status, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2016, amount feature over 2017, and amount payment in all years. PROJECT 3: SQLITE FOR DATA ANALYST AND DATA SCIENTIST WITH PYTHON GUI In this project, we will use the SQLite version of BikeStores database as a sample

database to help you work with MySQL quickly and effectively. The stores table includes the store's information. Each store has a store name, contact information such as phone and email, and an address including street, city, state, and zip code. The staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone. A staff works at a store specified by the value in the store_id column. A store can have one or more staffs. A staff reports to a store manager specified by the value in the manager_id column. If the value in the manager_id is null, then the staff is the top manager. If a staff no longer works for any stores, the value in the active column is set to zero. The categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes. The products table stores the product's information such as name, brand, category, model year, and list price. Each product belongs to a brand specified by the brand_id column. Hence, a brand may have zero or many products. Each product also belongs to a category specified by the category_id column. Also, each category may have zero or many products. The customers table stores customer's information including first name, last name, phone, email, street, city, state, zip code, and photo path. The orders table stores the sales order's header information including customer, order status, order date, required date, shipped date. It also stores the information on where the sales transaction was created (store) and who created it (staff). Each sales order has a row in the sales_orders table. A sales order has one or many line items stored in the order_items table. The order_items table stores the line items of a sales order. Each line item belongs to a sales order specified by the order_id column. A sales order line item includes product, order quantity, list price, and discount. The stocks table stores the inventory information i.e. the quantity of a particular product in a specific store. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by brand, top 10 sales by brand, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2017, amount feature over 2018, and all amount feature.

PROJECT 4: SQLITE FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI

In this project, you will use SQLite version of Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the SQLite database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997,

amount feature over 1998, and all amount feature. **PROJECT 5: ZERO TO MASTERY: THE COMPLETE GUIDE TO LEARNING SQLITE AND PYTHON GUI** In this project, we provide you with the SQLite version of The Oracle Database Sample Schemas that provides a common platform for examples in each release of the Oracle Database. The sample database is also a good database for practicing with SQL, especially SQLite. The detailed description of the database can be found on: <http://luna-ext.di.fc.ul.pt/oracle11g/server.112/e10831/diagrams.htm#insertedID0>. The four schemas are a set of interlinked schemas. This set of schemas provides a layered approach to complexity: A simple schema Human Resources (HR) is useful for introducing basic topics. An extension to this schema supports Oracle Internet Directory demos; A second schema, Order Entry (OE), is useful for dealing with matters of intermediate complexity. Many data types are available in this schema, including non-scalar data types; The Online Catalog (OC) subschema is a collection of object-relational database objects built inside the OE schema; The Product Media (PM) schema is dedicated to multimedia data types; The Sales History (SH) schema is designed to allow for demos with large amounts of data. An extension to this schema provides support for advanced analytic processing. The HR schema consists of seven tables: regions, countries, locations, departments, employees, jobs, and job_histories. This book only implements HR schema, since the other schemas will be implemented in the next books.

python gui for data analysis: MYSQL FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-06-26 In this project, you will use the Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called “Northwind Traders,” which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over june 1997, amount feature over 1998, and all amount feature.

python gui for data analysis: FOUR PROJECTS: MYSQL AND PYTHON GUI FOR DATA ANALYSIS Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-11-04 **PROJECT 1: FULL SOURCE CODE: MYSQL FOR STUDENTS AND PROGRAMMERS WITH PYTHON GUI** In this project, we provide you with a MySQL version of an Oracle sample database named OT which is based on a global fictitious company that sells computer hardware including storage, motherboard, RAM, video card, and CPU. The company maintains the product information such as name, description standard cost, list price, and product line. It also tracks the inventory information for all products including warehouses where products are available. Because the company operates globally, it has warehouses in various locations around the world. The company records all customer information including name, address, and website. Each customer has at least one contact person with detailed information including name, email, and phone. The company also places a credit limit on each customer to limit the amount that customer can owe. Whenever a customer issues a purchase order,

a sales order is created in the database with the pending status. When the company ships the order, the order status becomes shipped. In case the customer cancels an order, the order status becomes canceled. In addition to the sales information, the employee data is recorded with some basic information such as name, email, phone, job title, manager, and hire date. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by status, top 10 sales by status, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2016, amount feature over 2017, and amount payment in all years.

PROJECT 2: MYSQL FOR DATA ANALYST AND DATA SCIENTIST WITH PYTHON GUI In this project, we will use the BikeStores database as a MySQL sample database to help you work with MySQL quickly and effectively. The stores table includes the store's information. Each store has a store name, contact information such as phone and email, and an address including street, city, state, and zip code. The staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone. A staff works at a store specified by the value in the store_id column. A store can have one or more staffs. A staff reports to a store manager specified by the value in the manager_id column. If the value in the manager_id is null, then the staff is the top manager. If a staff no longer works for any stores, the value in the active column is set to zero. The categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes. The products table stores the product's information such as name, brand, category, model year, and list price. Each product belongs to a brand specified by the brand_id column. Hence, a brand may have zero or many products. Each product also belongs a category specified by the category_id column. Also, each category may have zero or many products. The customers table stores customer's information including first name, last name, phone, email, street, city, state, zip code, and photo path. The orders table stores the sales order's header information including customer, order status, order date, required date, shipped date. It also stores the information on where the sales transaction was created (store) and who created it (staff). Each sales order has a row in the sales_orders table. A sales order has one or many line items stored in the order_items table. The order_items table stores the line items of a sales order. Each line item belongs to a sales order specified by the order_id column. A sales order line item includes product, order quantity, list price, and discount. The stocks table stores the inventory information i.e. the quantity of a particular product in a specific store. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by category, top 10 sales by category, bottom 10 sales by brand, top 10 sales by brand, bottom 10 sales by customer city, top 10 sales by customer city, bottom 10 sales by customer state, top 10 sales by customer state, average amount by month with mean and EWM, average amount by every month, amount feature over June 2017, amount feature over 2018, and all amount feature.

PROJECT 3: MYSQL FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI In this project, you will use the Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry

accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over june 1997, amount feature over 1998, and all amount feature. PROJECT 4: MYSQL AND DATA SCIENCE: QUERIES AND VISUALIZATION WITH PYTHON GUI In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. This project uses the Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the MySQL from <https://dev.mysql.com/doc/sakila/en/>.

python gui for data analysis: FOUR PROJECTS: MySQL and SQLite For Data Science with Python GUI Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-06-29 PROJECT 1: SQLITE AND DATA SCIENCE: QUERIES AND VISUALIZATION WITH PYTHON GUI In this project, you will develop GUI with PyQt5 to: utilize Push Button, Combo Box, Table Widget, Line Edit, and Widget, read and create SQLite database and every table in it, plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. This project uses the Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the SQLite from <https://dev.mysql.com/doc/sakila/en/>. PROJECT 2: MYSQL AND DATA SCIENCE: QUERIES AND VISUALIZATION WITH PYTHON GUI In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with

PyQt5 to each table in the database. You will also create GUI to plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue costumers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005. This project uses the Sakila sample database which is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, film_actor, customer, rental, payment and inventory among others. You can download the MySQL from <https://dev.mysql.com/doc/sakila/en/>.

PROJECT 3: MYSQL FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI

In this project, you will use the Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature.

PROJECT 4: SQLITE FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI

In this project, you will use SQLite version of Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the SQLite database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour;

the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature.

python gui for data analysis: *Data Analysis and Visualization Using Python* Dr. Ossama Embarak, 2018-11-20 Look at Python from a data science point of view and learn proven techniques for data visualization as used in making critical business decisions. Starting with an introduction to data science with Python, you will take a closer look at the Python environment and get acquainted with editors such as Jupyter Notebook and Spyder. After going through a primer on Python programming, you will grasp fundamental Python programming techniques used in data science. Moving on to data visualization, you will see how it caters to modern business needs and forms a key factor in decision-making. You will also take a look at some popular data visualization libraries in Python. Shifting focus to data structures, you will learn the various aspects of data structures from a data science perspective. You will then work with file I/O and regular expressions in Python, followed by gathering and cleaning data. Moving on to exploring and analyzing data, you will look at advanced data structures in Python. Then, you will take a deep dive into data visualization techniques, going through a number of plotting systems in Python. In conclusion, you will complete a detailed case study, where you'll get a chance to revisit the concepts you've covered so far. What You Will Learn Use Python programming techniques for data science Master data collections in Python Create engaging visualizations for BI systems Deploy effective strategies for gathering and cleaning data Integrate the Seaborn and Matplotlib plotting systems Who This Book Is For Developers with basic Python programming knowledge looking to adopt key strategies for data analysis and visualizations using Python.

python gui for data analysis: DATA VISUALIZATION AND DATA ANALYTICS PROJECTS WITH MYSQL, SQLITE, POSTGRESQL, AND SQL SERVER USING PYTHON GUI Vivian Siahaan, Rismon Hasiholan Sianipar, 2022-10-26 **PROJECT 1: MYSQL FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI** In this project, you will use the Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature. **PROJECT 2: FULL SOURCE CODE: THE COMPLETE GUIDE TO LEARNING POSTGRESQL AND DATA SCIENCE WITH PYTHON GUI** In this project, we provide you with the PostgreSQL version of SQLite sample

database named chinook. The chinook sample database is a good database for practicing with SQL, especially PostgreSQL. The detailed description of the database can be found on: <https://www.sqlitetutorial.net/sqlite-sample-database/>. The sample database consists of 11 tables: The employee table stores employees data such as employee id, last name, first name, etc. It also has a field named ReportsTo to specify who reports to whom; customers table stores customers data; invoices & invoice_items tables: these two tables store invoice data. The invoice table stores invoice header data and the invoice_items table stores the invoice line items data; The artist table stores artists data. It is a simple table that contains only the artist id and name; The album table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums; The media_type table stores media types such as MPEG audio and AAC audio files; genre table stores music types such as rock, jazz, metal, etc; The track table stores the data of songs. Each track belongs to one album; playlist & playlist_track tables: The playlist table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship between the playlist table and track table is many-to-many. The playlist_track table is used to reflect this relationship. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, and day; the distribution of amount by year, quarter, month, week, day, and hour; the bottom/top 10 sales by employee, the bottom/top 10 sales by customer, the bottom/top 10 sales by customer, the bottom/top 10 sales by artist, the bottom/top 10 sales by genre, the bottom/top 10 sales by play list, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the bottom/top 10 sales by customer city, the payment amount by month with mean and EWM, the average payment amount by every month, and amount payment in all years.

PROJECT 3: FULL SOURCE CODE: SQL SERVER FOR DATA ANALYTICS AND VISUALIZATION WITH PYTHON GUI This book uses SQL SERVER version of MySQL-based Sakila sample database. It is a fictitious database designed to represent a DVD rental store. The tables of the database include film, film_category, actor, customer, rental, payment and inventory among others. The Sakila sample database is intended to provide a standard schema that can be used for examples in books, tutorials, articles, samples, and so forth. Detailed information about the database can be found on website: <https://dev.mysql.com/doc/index-other.html>. In this project, you will develop GUI using PyQt5 to: read SQL SERVER database and every table in it; read every actor in actor table, read every film in films table; plot case distribution of film release year, film rating, rental duration, and categorize film length; plot rating variable against rental_duration variable in stacked bar plots; plot length variable against rental_duration variable in stacked bar plots; read payment table; plot case distribution of Year, Day, Month, Week, and Quarter of payment; plot which year, month, week, days of week, and quarter have most payment amount; read film list by joining five tables: category, film_category, film_actor, film, and actor; plot case distribution of top 10 and bottom 10 actors; plot which film title have least and most sales; plot which actor have least and most sales; plot which film category have least and most sales; plot case distribution of top 10 and bottom 10 overdue customers; plot which customer have least and most overdue days; plot which store have most sales; plot average payment amount by month with mean and EWM; and plot payment amount over June 2005.

PROJECT 4: SQLITE FOR DATA ANALYSIS AND VISUALIZATION WITH PYTHON GUI In this project, you will use SQLite version of Northwind database which is a sample database that was originally created by Microsoft and used as the basis for their tutorials in a variety of database products for decades. The Northwind database contains the sales data for a fictitious company called "Northwind Traders," which imports and exports specialty foods from around the world. The Northwind database is an excellent tutorial schema for a small-business ERP, with customers, orders, inventory, purchasing, suppliers, shipping, employees, and single-entry accounting. The Northwind dataset includes sample data for the following: Suppliers: Suppliers and vendors of Northwind; Customers: Customers who buy products from Northwind; Employees: Employee details of Northwind traders; Products: Product information; Shippers: The details of the shippers who ship

the products from the traders to the end-customers; Orders and Order_Details: Sales Order transactions taking place between the customers & the company. The Northwind sample database includes 11 tables and the table relationships are showcased in the following entity relationship diagram. In this project, you will write Python script to create every table and insert rows of data into each of them. You will develop GUI with PyQt5 to each table in the SQLite database. You will also create GUI to plot: case distribution of order date by year, quarter, month, week, day, and hour; the distribution of amount by year, quarter, month, week, day, and hour; the distribution of bottom 10 sales by product, top 10 sales by product, bottom 10 sales by customer, top 10 sales by customer, bottom 10 sales by supplier, top 10 sales by supplier, bottom 10 sales by customer country, top 10 sales by customer country, bottom 10 sales by supplier country, top 10 sales by supplier country, average amount by month with mean and ewm, average amount by every month, amount feature over June 1997, amount feature over 1998, and all amount feature.

python gui for data analysis: Data Analytics Essentials You Always Wanted To Know Vibrant Publishers, Dr. Bianca Szasz, 2024-02-29 Upon reading this book, you will get: □ A fundamental comprehension of data analytics, including its types □ An understanding of data analytics processes, software tools, and a range of analytics methodologies □ A comprehension of what daily tasks and procedures the data analysts follow □ An investigation into the vast field of big data analytics, covering its possibilities and challenges □ An understanding of the existing legal frameworks, as well as ethical and privacy issues in data analytics □ Application-based learning using a variety of real-world case studies From raw data to actionable insights - journey through the essentials of data analytics. Data Analytics Essentials You Always Wanted To Know is an approachable and captivating guide to understand the complicated world of data Data analytics is becoming increasingly important in today's data-driven society, and so has the demand for data analysts. Data Analytics Essentials You Always Wanted to Know (Data Analytics Essentials) is a comprehensive yet succinct manual, perfect for you if you are trying to understand the fundamentals of data analytics. It gives a concise introduction to data analytics and its current applicability. This book is a great tool for professionals switching to a career in data analytics and for students who want to learn the basics of data analytics. It will give you a strong foundation by explaining everything in an easy-to-understand language. Data Analytics Essentials goes beyond a theoretical manual and contains real-world case studies and fun facts to help you enhance your knowledge. The chapter summaries and self- assessment tests along with every chapter will help you test yourself as you move from one concept to the next.

python gui for data analysis: Big Data Analytics and Knowledge Discovery Robert Wrembel, Silvia Chiusano, Gabriele Kotsis, A Min Tjoa, Ismail Khalil, 2024-08-17 This book constitutes the proceedings of the 26th International Conference on Big Data Analytics and Knowledge Discovery, DaWaK 2024, which too place in Naples, Italy, during August 26-28, 2024. The 16 full and 20 short papers included in this book were carefully reviewed and selected from 83 submissions. They were organized in topical sections as follows: Modeling and design; entity matching and similarity; classification; machine learning methods and applications; time series; data repositories; optimization; and data quality and applications.

python gui for data analysis: Python: Data Analytics and Visualization Phuong Vo.T.H, Martin Czygan, Ashish Kumar, Kirthi Raman, 2017-03-31 Understand, evaluate, and visualize data About This Book Learn basic steps of data analysis and how to use Python and its packages A step-by-step guide to predictive modeling including tips, tricks, and best practices Effectively visualize a broad set of analyzed data and generate effective results Who This Book Is For This book is for Python Developers who are keen to get into data analysis and wish to visualize their analyzed data in a more efficient and insightful manner. What You Will Learn Get acquainted with NumPy and use arrays and array-oriented computing in data analysis Process and analyze data using the time-series capabilities of Pandas Understand the statistical and mathematical concepts behind predictive analytics algorithms Data visualization with Matplotlib Interactive plotting with NumPy, Scipy, and MKL functions Build financial models using Monte-Carlo simulations Create directed graphs and

multi-graphs Advanced visualization with D3 In Detail You will start the course with an introduction to the principles of data analysis and supported libraries, along with NumPy basics for statistics and data processing. Next, you will overview the Pandas package and use its powerful features to solve data-processing problems. Moving on, you will get a brief overview of the Matplotlib API .Next, you will learn to manipulate time and data structures, and load and store data in a file or database using Python packages. You will learn how to apply powerful packages in Python to process raw data into pure and helpful data using examples. You will also get a brief overview of machine learning algorithms, that is, applying data analysis results to make decisions or building helpful products such as recommendations and predictions using Scikit-learn. After this, you will move on to a data analytics specialization—predictive analytics. Social media and IOT have resulted in an avalanche of data. You will get started with predictive analytics using Python. You will see how to create predictive models from data. You will get balanced information on statistical and mathematical concepts, and implement them in Python using libraries such as Pandas, scikit-learn, and NumPy. You'll learn more about the best predictive modeling algorithms such as Linear Regression, Decision Tree, and Logistic Regression. Finally, you will master best practices in predictive modeling. After this, you will get all the practical guidance you need to help you on the journey to effective data visualization. Starting with a chapter on data frameworks, which explains the transformation of data into information and eventually knowledge, this path subsequently cover the complete visualization process using the most popular Python libraries with working examples This Learning Path combines some of the best that Packt has to offer in one complete, curated package. It includes content from the following Packt products: Getting Started with Python Data Analysis, Phuong Vo.T.H & Martin Czygan Learning Predictive Analytics with Python, Ashish Kumar Mastering Python Data Visualization, Kirthi Raman Style and approach The course acts as a step-by-step guide to get you familiar with data analysis and the libraries supported by Python with the help of real-world examples and datasets. It also helps you gain practical insights into predictive modeling by implementing predictive-analytics algorithms on public datasets with Python. The course offers a wealth of practical guidance to help you on this journey to data visualization

python gui for data analysis: Statistical Genomics Brooke Fridley, Xuefeng Wang, 2023-03-16 This volume provides a collection of protocols from researchers in the statistical genomics field. Chapters focus on integrating genomics with other “omics” data, such as transcriptomics, epigenomics, proteomics, metabolomics, and metagenomics. Written in the highly successful Methods in Molecular Biology series format, chapters include introductions to their respective topics, lists of the necessary materials and reagents, step-by-step, readily reproducible laboratory protocols, and tips on troubleshooting and avoiding known pitfalls. Cutting-edge and thorough, Statistical Genomics hopes that by covering these diverse and timely topics researchers are provided insights into future directions and priorities of pan-omics and the precision medicine era.

python gui for data analysis: Data Analytics using Python Mahmoud Ahmad Al-Khasawneh, 2024-08-07 Descriptive statistics are used to identify the fundamental characteristics of data in a research study. Simply summarized information about the sample and measurements is provided. Descriptive statistics provide information about the components and dissemination of values in single or multiple data set concisely. The classical illustrative statistics allow experts to get a quick sense of the central inclination and degree of diffusion of values in a dataset with a single glance. They are useful in gaining an understanding of data distribution as well as in comparing different data distributions, It is frequently necessary for human geographers to take into consideration the locational citations of the data they are working with. Using spatial descriptive statistics, analysts can determine the central propensity and variation of data in a given geographic area or region. The two types of illustrative analysis are mutually supportive of one another. Experts can research the geographic phenomena with which they are involved by combining both statistics and mathematics. Even though descriptive statistics are straightforward concepts in statistical assessment, they are essential and beneficial in today's world of massive amounts of data. The performance and efficacy of descriptive analysis should not be overshadowed in the face of ever-increasing huge quantities of

data being generated continuously and distributed via the Internet. Descriptive statistics are characterized by inferential analysis in most cases. When you use descriptive statistics, you are merely explaining what is or what the information reveals about something. When using inferential analysis, you are attempting to draw conclusions that are not based solely on the available data. For example, we use inferential analysis to try to infer what the general public might think based on a sample of data. Alternatively, we use inferential analysis to make decisions about the likelihood that a difference between groups observed in this study is a dependable difference or one that could have occurred by chance. As a result, we use inferential analysis to conclude more general conditions from our data, whereas we use descriptive analysis to simply describe what is happening in our data. Descriptive statistics are used to present quantitative explanations in a manageable format. In a research study, we may have a large number of measures. Alternatively, we can quantify a huge number of participants using any measure.

python gui for data analysis: Data Science and Analytics Sneha Kumari, K.K. Tripathy, Vidya Kumbhar, 2020-12-04 Data Science and Analytics explores the application of big data and business analytics by academics, researchers, industrial experts, policy makers and practitioners, helping the reader to understand how big data can be efficiently utilized in better managerial applications.

python gui for data analysis: TKINTER, DATA SCIENCE, AND MACHINE LEARNING Vivian Siahaan, Rismon Hasiholan Sianipar, 2023-09-02 In this project, we embarked on a comprehensive journey through the world of machine learning and model evaluation. Our primary goal was to develop a Tkinter GUI and assess various machine learning models on a given dataset to identify the best-performing one. This process is essential in solving real-world problems, as it helps us select the most suitable algorithm for a specific task. By crafting this Tkinter-powered GUI, we provided an accessible and user-friendly interface for users engaging with machine learning models. It simplified intricate processes, allowing users to load data, select models, initiate training, and visualize results without necessitating code expertise or command-line operations. This GUI introduced a higher degree of usability and accessibility to the machine learning workflow, accommodating users with diverse levels of technical proficiency. We began by loading and preprocessing the dataset, a fundamental step in any machine learning project. Proper data preprocessing involves tasks such as handling missing values, encoding categorical features, and scaling numerical attributes. These operations ensure that the data is in a format suitable for training and testing machine learning models. Once our data was ready, we moved on to the model selection phase. We evaluated multiple machine learning algorithms, each with its strengths and weaknesses. The models we explored included Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Decision Trees, Gradient Boosting, Extreme Gradient Boosting (XGBoost), Multi-Layer Perceptron (MLP), and Support Vector Classifier (SVC). For each model, we employed a systematic approach to find the best hyperparameters using grid search with cross-validation. This technique allowed us to explore different combinations of hyperparameters and select the configuration that yielded the highest accuracy on the training data. These hyperparameters included settings like the number of estimators, learning rate, and kernel function, depending on the specific model. After obtaining the best hyperparameters for each model, we trained them on our preprocessed dataset. This training process involved using the training data to teach the model to make predictions on new, unseen examples. Once trained, the models were ready for evaluation. We assessed the performance of each model using a set of well-established evaluation metrics. These metrics included accuracy, precision, recall, and F1-score. Accuracy measured the overall correctness of predictions, while precision quantified the proportion of true positive predictions out of all positive predictions. Recall, on the other hand, represented the proportion of true positive predictions out of all actual positives, highlighting a model's ability to identify positive cases. The F1-score combined precision and recall into a single metric, helping us gauge the overall balance between these two aspects. To visualize the model's performance, we created key graphical representations. These included confusion matrices, which showed the number of true positive, true

negative, false positive, and false negative predictions, aiding in understanding the model's classification results. Additionally, we generated Receiver Operating Characteristic (ROC) curves and area under the curve (AUC) scores, which depicted a model's ability to distinguish between classes. High AUC values indicated excellent model performance. Furthermore, we constructed true values versus predicted values diagrams to provide insights into how well our models aligned with the actual data distribution. Learning curves were also generated to observe a model's performance as a function of training data size, helping us assess whether the model was overfitting or underfitting. Lastly, we presented the results in a clear and organized manner, saving them to Excel files for easy reference. This allowed us to compare the performance of different models and make an informed choice about which one to select for our specific task. In summary, this project was a comprehensive exploration of the machine learning model development and evaluation process. We prepared the data, selected and fine-tuned various models, assessed their performance using multiple metrics and visualizations, and ultimately arrived at a well-informed decision about the most suitable model for our dataset. This approach serves as a valuable blueprint for tackling real-world machine learning challenges effectively.

python gui for data analysis: DATA VISUALIZATION, TIME-SERIES FORECASTING, AND PREDICTION USING MACHINE LEARNING WITH TKINTER Vivian Siahaan, Rismon Hasiholan Sianipar, 2023-09-06 This Data Visualization, Time-Series Forecasting, and Prediction using Machine Learning with Tkinter project is a comprehensive and multifaceted application that leverages data visualization, time-series forecasting, and machine learning techniques to gain insights into bitcoin data and make predictions. This project serves as a valuable tool for financial analysts, traders, and investors seeking to make informed decisions in the stock market. The project begins with data visualization, where historical bitcoin market data is visually represented using various plots and charts. This provides users with an intuitive understanding of the data's trends, patterns, and fluctuations. Features distribution analysis is conducted to assess the statistical properties of the dataset, helping users identify key characteristics that may impact forecasting and prediction. One of the project's core functionalities is time-series forecasting. Through a user-friendly interface built with Tkinter, users can select a stock symbol and specify the time horizon for forecasting. The project supports multiple machine learning regressors, such as Linear Regression, Decision Trees, Random Forests, Gradient Boosting, Extreme Gradient Boosting, Multi-Layer Perceptron, Lasso, Ridge, AdaBoost, and KNN, allowing users to choose the most suitable algorithm for their forecasting needs. Time-series forecasting is crucial for making predictions about stock prices, which is essential for investment strategies. The project employs various machine learning regressors to predict the adjusted closing price of bitcoin stock. By training these models on historical data, users can obtain predictions for future adjusted closing prices. This information is invaluable for traders and investors looking to make buy or sell decisions. The project also incorporates hyperparameter tuning and cross-validation to enhance the accuracy of these predictions. These models employ metrics such as Mean Absolute Error (MAE), which quantifies the average absolute discrepancy between predicted values and actual values. Lower MAE values signify superior model performance. Additionally, Mean Squared Error (MSE) is used to calculate the average squared differences between predicted and actual values, with lower MSE values indicating better model performance. Root Mean Squared Error (RMSE), derived from MSE, provides insights in the same units as the target variable and is valued for its lower values, denoting superior performance. Lastly, R-squared (R^2) evaluates the fraction of variance in the target variable that can be predicted from independent variables, with higher values signifying better model fit. An R^2 of 1 implies a perfect model fit. In addition to close price forecasting, the project extends its capabilities to predict daily returns. By implementing grid search, users can fine-tune the hyperparameters of machine learning models such as Random Forests, Gradient Boosting, Support Vector, Decision Tree, Gradient Boosting, Extreme Gradient Boosting, Multi-Layer Perceptron, and AdaBoost Classifiers. This optimization process aims to maximize the predictive accuracy of daily returns. Accurate daily return predictions are essential for assessing risk and formulating effective

trading strategies. Key metrics in these classifiers encompass Accuracy, which represents the ratio of correctly predicted instances to the total number of instances, Precision, which measures the proportion of true positive predictions among all positive predictions, and Recall (also known as Sensitivity or True Positive Rate), which assesses the proportion of true positive predictions among all actual positive instances. The F1-Score serves as the harmonic mean of Precision and Recall, offering a balanced evaluation, especially when considering the trade-off between false positives and false negatives. The ROC Curve illustrates the trade-off between Recall and False Positive Rate, while the Area Under the ROC Curve (AUC-ROC) summarizes this trade-off. The Confusion Matrix provides a comprehensive view of classifier performance by detailing true positives, true negatives, false positives, and false negatives, facilitating the computation of various metrics like accuracy, precision, and recall. The selection of these metrics hinges on the project's specific objectives and the characteristics of the dataset, ensuring alignment with the intended goals and the ramifications of false positives and false negatives, which hold particular significance in financial contexts where decisions can have profound consequences. Overall, the Data Visualization, Time-Series Forecasting, and Prediction using Machine Learning with Tkinter project serves as a powerful and user-friendly platform for financial data analysis and decision-making. It bridges the gap between complex machine learning techniques and accessible user interfaces, making financial analysis and prediction more accessible to a broader audience. With its comprehensive features, this project empowers users to gain insights from historical data, make informed investment decisions, and develop effective trading strategies in the dynamic world of finance. You can download the dataset from: <http://viviansiahaan.blogspot.com/2023/09/data-visualization-time-series.html>.

python gui for data analysis: Knowledge Discovery in Databases: PKDD 2004

Jean-Francois Boulicaut, Floriana Esposito, Fosca Giannotti, Dino Pedreschi, 2004-09-10 This book constitutes the refereed proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2004, held in Pisa, Italy, in September 2004 jointly with ECML 2004. The 39 revised full papers and 9 revised short papers presented together with abstracts of 5 invited talks were carefully reviewed and selected from 194 papers submitted to PKDD and 107 papers submitted to both, PKDD and ECML. The papers present a wealth of new results in knowledge discovery in databases and address all current issues in the area.

python gui for data analysis: Modern PyQt Joshua Willman, 2020-12-29 Dive into GUI application development and create useful applications for practical and relevant topics in the fields of business, computer science, and research. This book uses a realistic approach to help get you started designing and building the applications you need while learning new tools along the way. PyQt has a vast collection of tools that you can use to create GUIs, many of which seem to go unexplored. In Modern PyQt, you will go beyond some of the fundamental topics of GUI development in order to begin building useful desktop applications. Through extensive examples and hands-on projects, you will explore how to make applications for data analysis and visualization using graphs, computer vision with OpenCV and PyQt, the basics of networking, handling databases with SQL, and more! Whether you are looking for new ideas to practice your skills as a programmer or you have a specific goal in mind and need some help to get your ideas off the ground, there is something in Modern PyQt for you! What You Will Learn Create cross-platform GUIs with Python and PyQt. Understand the important PyQt classes, widgets, and concepts needed for building interactive and practical applications. Find out how to embed useful Python modules into your applications to create more advanced GUIs. Build useful applications that you can improve or make into something completely new with Python and PyQt. Who This Book Is For Intermediate level programmers or above in Python. GUI developers with some experience designing GUIs. Even if they have never used PyQt before, the concepts learned from other toolkits, such as Tkinter or wxPython, can be carried over for developing applications with using PyQt.

python gui for data analysis: BUSINESS STATISTICS & ANALYTICS FOR DECISION

MAKING: Made Simple Dr. Mukul Burghate I Dr. Padmakar Shahare, The analysis of statistics in business for better decision making is nowadays called Big Data Analytics. Big data analytics refers

to the process of collecting, organizing and analyzing large sets of data (called big data) to discover patterns and other useful information. Big data analytics can help organizations to better understand the information contained within the data and will also help identify the data that is most important to the business and future business decisions. Analysts working with big data basically want the knowledge that comes from analyzing the data. The purpose of this textbook is to present an introduction to the BUSINESS STATISTICS & ANALYTICS FOR DECISION MAKING subject of Management & Commerce. The book contains the syllabus from basics of the subjects going into the intricacies of the subjects. All the concepts have been explained with relevant Numerals, examples and diagrams to make it interesting for the readers. An attempt is made here by the experts to assist the students by way of providing Study Material as per the curriculum with non-commercial considerations. However, it is implicit that these are exam-oriented Study Material and students are advised to attend regular lectures in the Institute and utilize reference books available in the library for In-depth knowledge. We owe to many websites and their free contents; we would like to specially acknowledge contents of website www.wikipedia.com and various authors whose writings formed the basis for this book. We acknowledge our thanks to them. At the end we would like to say that there is always a room for improvement in whatever we do. We would appreciate any suggestions regarding this study material from the readers so that the contents can be made more interesting and meaningful. Readers can email their queries and doubts to our authors on tmcnagpur@gmail.com. We shall be glad to help you immediately. Authors: Dr Mukul Burghate and Dr Padmakar Shahare

python gui for data analysis: *Machine Learning and Big Data Analytics* Rajiv Misra, Rana Omer, Muttukrishnan Rajarajan, Bharadwaj Veeravalli, Nishtha Kesswani, Priyanka Mishra, 2023-06-06 This edited volume on machine learning and big data analytics (Proceedings of ICMLBDA 2022) is intended to be used as a reference book for researchers and professionals to share their research and reports of new technologies and applications in Machine Learning and Big Data Analytics like biometric Recognition Systems, medical diagnosis, industries, telecommunications, AI Petri Nets Model-Based Diagnosis, gaming, stock trading, Intelligent Aerospace Systems, robot control, law, remote sensing and scientific discovery agents and multiagent systems; and natural language and Web intelligence. The intent of this book is to provide awareness of algorithms used for machine learning and big data in the advanced Scientific Technologies, provide a correlation of multidisciplinary areas and become a point of great interest for Data Scientists, systems architects, developers, new researchers and graduate level students. This volume provides cutting-edge research from around the globe on this field. Current status, trends, future directions, opportunities, etc. are discussed, making it friendly for beginners and young researchers.

Related to python gui for data analysis

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Download Python | Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands as a successor of a language called ABC. Guido remains

Python's principal

Python Tutorial - W3Schools Well organized and easy to understand Web building tutorials with lots of examples of how to use HTML, CSS, JavaScript, SQL, Python, PHP, Bootstrap, Java, XML and more

Python Tutorials - Real Python Learn Python online: Python tutorials for developers of all skill levels, Python books and courses, Python news, code examples, articles, and more

Learn Python Programming Language - GeeksforGeeks In this section, we'll cover the basics of Python programming, including installing Python, writing first program, understanding comments and working with variables, keywords

Learn Python - Free Interactive Python Tutorial Get started learning Python with DataCamp's free Intro to Python tutorial. Learn Data Science by completing interactive coding challenges and watching videos by expert instructors

Welcome to Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. Whet your appetite with our **Python Basics - Real Python** On this page you'll find fundamental concepts for Python beginners that will help you get started on your journey to learn Python. These tutorials focus on the absolutely essential

Related to python gui for data analysis

Automating Data Analysis with Python Dashboards (The CPA Journal14d) In today's data-rich environment, business are always looking for a way to capitalize on available data for new insights and

Automating Data Analysis with Python Dashboards (The CPA Journal14d) In today's data-rich environment, business are always looking for a way to capitalize on available data for new insights and

Creating Python GUIs With GIMP (Hackaday2d) GUI design can be a tedious job, requiring the use of specialist design tools and finding a suitable library that fits your

Creating Python GUIs With GIMP (Hackaday2d) GUI design can be a tedious job, requiring the use of specialist design tools and finding a suitable library that fits your

How to Use pandas DataFrames in Python to Analyze and Manipulate Data (How-To Geek on MSN4d) pandas is a Python module that's popular in data science and data analysis. It's offers a way to organize data into

How to Use pandas DataFrames in Python to Analyze and Manipulate Data (How-To Geek on MSN4d) pandas is a Python module that's popular in data science and data analysis. It's offers a way to organize data into

Quick Excel Python Hacks for Advanced Automated Data Analysis (Geeky Gadgets3mon) What if you could turn Excel into a powerhouse for advanced data analysis and automation in just a few clicks? Imagine effortlessly cleaning messy datasets, running complex calculations, or generating

Quick Excel Python Hacks for Advanced Automated Data Analysis (Geeky Gadgets3mon) What if you could turn Excel into a powerhouse for advanced data analysis and automation in just a few clicks? Imagine effortlessly cleaning messy datasets, running complex calculations, or generating

Databot: AI-assisted data analysis in R or Python (InfoWorld27d) If you'd like an LLM to act more like a partner than a tool, Databot is an experimental alternative to querychat that also works in both R and Python. Databot is designed to analyze data you've

Databot: AI-assisted data analysis in R or Python (InfoWorld27d) If you'd like an LLM to act more like a partner than a tool, Databot is an experimental alternative to querychat that also works in both R and Python. Databot is designed to analyze data you've

Transitioning From Excel To Python: Essential Functions For SEO Data Analysis

(Searchenginejournal.com2y) Still using Excel for your data analysis? Learn how to leverage Python so you can work with larger datasets and automate repetitive tasks. Learning to code, whether with Python, JavaScript, or another

Transitioning From Excel To Python: Essential Functions For SEO Data Analysis

(Searchenginejournal.com2y) Still using Excel for your data analysis? Learn how to leverage Python so you can work with larger datasets and automate repetitive tasks. Learning to code, whether with Python, JavaScript, or another

Related Articles

- [common core fourth grade math standards](#)
- [how to win friends and influence people course](#)
- [doctor of biblical studies](#)

[Back to Home](#)