CAREERFOUNDRY

Data Visualization with Python: Creating a Strategic Dashboard

Objective

Imagine you're the lead analyst for a bike-sharing service based in New York City, USA. Your team has been tasked with analyzing user behavior to help the business strategy department assess the current logistics model of bike distribution across the city and identify expansion opportunities.

The project's objective is to conduct a descriptive analysis of existing data and discover actionable insights for the business strategy team to help make informed decisions that will circumvent availability issues and ensure the company's position as a leader in eco-friendly transportation solutions in the city.

Context

For this project, you'll be using public data from New York bike-sharing facilities operated by Citi Bike. For context, Citi Bike's popularity has increased since its launch in 2013. The company's marketing strategy promotes bike sharing as a sustainable and convenient means of transportation, which has been very successful. Since the Covid–19 pandemic, New York residents have found even more merit in bike sharing, creating higher demand. This has led to distribution problems—such as fewer bikes at popular bike stations or stations full of docked bikes, making it difficult to return a hired bike—and customer complaints.

As the lead analyst, your task is to diagnose where distribution issues stem from and advise higher management on a solution based on your diagnosis of the root of the problem—whether it's sheer numbers, seasonal demand, or something else. Being in a management position also makes you the bridge between divisions, which requires you to ensure the information is tangible for the business development team. To effectively communicate your analysis to non-analysts, you'll present your insights in an interactive dashboard depicting the metrics you identify as vital for tackling the distribution issues.

Data and Tool Requirements

For this project, you'll use open source data from the Citi Bike database for the year 2022. To enrich this data set, you'll gather weather data using NOAA's API service. You will then use Python libraries, including Matplotlib, Seaborn, and Plotly to make charts; pandas and Kepler.gl, to create maps; and Streamlit to design your final dashboard.

Data Sourcing Criteria

- Download an open source data set from Citi Bike for the year 2022;
- Gather weather data for the year 2022 using NOAA's API.

Analysis Criteria

- Use descriptive analysis and apply aggregations of bike trips across New York to discover the most popular starting locations and summarize data yearly to find seasonal patterns.
- Apply geographic plotting to identify problem areas in station distribution and explore the most common routes.

Visualization Criteria

- Create a dashboard in Python to showcase the findings of your analysis. The dashboard should be divided into sections based on the variables affecting demand.
- Each graph should be visually appealing and refined using a color palette in Python.
- The dashboard should contain an introduction to set up the presentation, a slide for each graph to avoid overwhelming the audience, and a recommendations page that clearly explains the outcomes of the analysis.

Success Factors

- Present an interactive dashboard that depicts all the problematic aspects of bike logistics around the city.
- Your dashboard contains visualizations that seek to understand the different factors that could affect the business.
- You present strategy recommendations and next steps based on your findings.

Project Deliverables

Throughout this Achievement, you'll work from Exercise to Exercise to complete your project. For each Exercise task, you'll submit a deliverable that directly contributes to the final product—in this case, a network visualization.

Here is a breakdown of your project deliverables by Exercise.

Exercise 2.1 Tool for Creating Dashboards

- Analyze different visualization tools by comparing their usability in different scenarios;
- Analyze the different Python libraries available for dashboards by conducting research;
- Evaluate a Python dashboard.

Exercise 2.2 Sourcing Web Data with an API

- Analyze a task by creating a list of questions to solve a problem;
- Develop a plan to create a dashboard;
- Apply an API to a web page to gather data;
- Apply Python code to structured and unstructured data sets to clean and merge them.

Exercise 2.3 Fundamentals of Visualizations Part 1

- Differentiate between procedural and object-oriented programming;
- Create bar and line charts using matplotlib;
- Define components in visualization charts by applying customization features;
- Analyze visualizations to interpret data.

Exercise 2.4 Fundamentals of Visualizations Part 2

- Apply matplotlib principles to seaborn;
- Create bar and line charts using seaborn;
- Analyze a categorical variable using a visualization;
- Explain the benefits of utilizing a FacetGrid for analysis.

Exercise 2.5 Advanced Geospatial Plotting

- Apply functions in Python to aggregate data;
- Create a geospatial plot using kepler.gl;
- Analyze a map by applying filtering to recognize patterns in data.

Exercise 2.6 Creating a Python Dashboard

- Create charts in Plotly;
- Execute a Python script to run a program;
- Apply code to initialize a Python dashboard in Streamlit;
- Create a dashboard integrating multiple Python libraries.

Exercise 2.7 Refining and Presenting a Python Dashboard

- Apply fundamental design principles to a dashboard;
- Create a multi-page dashboard using Streamlit;
- Deploy a dashboard using Streamlit;
- Analyze data by interpreting visualizations;
- Deliver an analysis through a presentation.