Session 2.2: Business Data Visualization: Exploratory Plotting and Executive Dashboards

THE BUSINESS SITUATION

- The marketing team keeps 12 months of results in one table: revenue, marketing spend, cost per new customer, and customer loyalty
- Executives often have only a minute or two per chart, and a table of 72 numbers hides the trend
- The CMO (Chief Marketing Officer) asks: is our marketing spend paying off, and are customers getting happier? Today, Python turns the table into charts that answer this at a glance

Today's flow: 60 min lecture, then 120 min hands-on lab

EXPECTED OUTCOMES

By the end of today, you will be able to:

1. ***"Is revenue growing month by month?"***
   Draw a line chart with Matplotlib (Python's main charting library) using `fig, ax = plt.subplots()` (`fig` is the whole image; `ax` is one chart inside it)
2. ***"In which months did we spend the most on marketing?"***
   Draw a bar chart with Seaborn (a library built on Matplotlib for quick, clean statistical charts)
3. ***"Which numbers move together?"***
   Read a correlation heatmap (a colored grid of correlations, each from -1 to +1), and fix an `IndexError` in a 2x2 dashboard

Session Map: Follow the Numbers

Today's question: *Is our marketing spend paying off, and are customers getting happier?*

1 Why it matters
Lecture, part 1

1. **Session Purpose**
   answer today's question
2. **Learning Outcomes**
   3 skills you will show today
3. **Business Context**
   A Chart Answers One Question
4. **Data Basics**
   Pick the Chart from the Question

→

2 Learn the tools
Lecture, part 2

5. **Lab Dataset**
   1 CSV
6. **Keywords**
   6 terms
7. **Core Foundations**
   6 ideas, one at a time
8. **IDE Setup**
   `pip install matplotlib seaborn`
9. **Code Examples**
   3 demos + Quick Checks
10. **References**
    3 official docs

→

3 Practice
Lab

11. **Exercise 1: Guided**
    Two-Panel Executive Dashboard
12. **Exercise 2: Applied**
    Four-Panel Boardroom Dashboard
13. **Exercise 3: Advanced**
    Is Marketing Getting Less Efficient?

A Chart Answers One Question

KEY POINTS

- A table makes the reader hunt for the pattern; a chart shows it.
- Five rules for charts that executives trust:
  - One message per chart, and the title states it, for example "Revenue More Than Doubled from Jan to Dec"
  - Label every axis with its unit, such as "USD M" (millions of U.S. Dollars)
  - Bar charts start at zero; a cut-off axis makes small differences look huge
  - No 3D effects or decoration; every mark on the chart should carry data
  - Use colors that still work in black-and-white print and for color-blind readers

Pick the Chart from the Question

KEY POINTS

- **Correlation** measures how closely two columns move together, on a scale from -1 to +1:
  - `+1`: when one goes up, the other always goes up
  - `0`: no straight-line pattern
  - `-1`: when one goes up, the other always goes down
- Correlation is not causation.

AT A GLANCE

Business question	Chart	Python
How does a number change over time?	Line chart	`ax.plot()`
Which category is biggest?	Bar chart	`sns.barplot()` or `ax.bar()`
Do two numbers move together?	Scatter plot	`ax.scatter()`
Which of many numbers move together?	Correlation heatmap	`sns.heatmap()`

The Data for Today

FILES

- `data/session_02_02_dataset.csv`

Audited 12-row monthly performance dataset (fictional teaching data), one row per month:

COLUMNS

`Month_Index`	Month number, 1 to 12.
`Month_Name`	Three-letter month name, `Jan` through `Dec`.
`Organic_Revenue_USD_M`	Organic revenue in millions of U.S. Dollars. "Organic" means revenue from the company's own operations, not from buying other companies.
`Paid_Marketing_Spend_USD_M`	Paid advertising spend in millions of U.S. Dollars.
`Customer_Acquisition_Cost_USD`	CAC (Customer Acquisition Cost): the average marketing cost to win one new customer, in U.S. Dollars.
`Net_Promoter_Score`	NPS (Net Promoter Score): a customer loyalty score from -100 to +100, based on how likely customers are to recommend the company. Our values range from `58` to `74`.

6 Terms You Need Today

TERM 1

`Matplotlib` and `Seaborn`

What: Python charting libraries; Seaborn adds quick, clean statistical charts on top of Matplotlib

Why: they turn a DataFrame into a chart image with a few lines of code

TERM 2

`Figure` and `Axes`

What: the `Figure` is the whole image; each `Axes` is one chart inside it

Why: you draw on a specific `Axes`, so a dashboard can hold several charts

TERM 3

`Line Chart`

What: points joined in time order

Why: shows trends, such as monthly revenue

TERM 4

`Bar Chart`

What: one bar per category; the bar length is the value

Why: compares sizes, such as spend by month

TERM 5

`Scatter Plot`

What: one dot per row, with one column on each axis

Why: shows whether two numbers move together

TERM 6

`Correlation Heatmap`

What: a grid of correlations, colored from -1 to +1

Why: spots which of many numbers move together, in one picture

How It Works

7.1

Figure and Axes

Remember: the Figure is the page; each Axes is a chart; draw on a named `ax`.

7.2

Choose the chart from the question

Remember: question first, chart second.

7.3

Correlation

Remember: correlation measures moving together, not cause and effect.

7.4

Saving a chart

Remember: inches x dpi = pixels; save before you show.

7.5

Grids of charts

Remember: `axes[row, column]`, both from `0`.

7.6

Honest charts

Remember: if the axis is cut, the bar chart lies.

7.1 Figure and Axes: On Real Data

SEE IT ON REAL DATA

Instruction	Goes to	Effect
`fig, ax = plt.subplots(figsize=(10, 5))`	creates both	a 10 x 5 inch image with one chart area
`ax.plot(df["Month_Name"], df["Organic_Revenue_USD_M"])`	the chart	draws 12 points joined by a line
`ax.set_title("Revenue More Than Doubled from Jan to Dec")`	the chart	the chart's title
`ax.set_ylabel("Organic Revenue (USD M)")`	the chart	the y-axis label
`fig.savefig("data/revenue_trend.png")`	the whole image	writes the image to a file

Remember: the Figure is the page; each Axes is a chart; draw on a named `ax`.

Line Chart: Revenue Trend

YourName_S22_demo_1.py — Input → Process → Output

```
# 1. Input: load the monthly results
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("data/session_02_02_dataset.csv")

# 2. Process: one figure with one chart, revenue by month
fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(df["Month_Name"], df["Organic_Revenue_USD_M"],
        color="#991B1B", marker="o")
ax.set_title("Revenue More Than Doubled from Jan to Dec")
ax.set_ylabel("Organic Revenue (USD M)")

# 3. Output: save the chart as an image file
fig.savefig("data/revenue_trend.png", dpi=150, bbox_inches="tight")
print("Saved: data/revenue_trend.png")
```

HOW IT WORKS

- First time only, in the Terminal: `pip install matplotlib seaborn`
- `import matplotlib.pyplot as plt` loads Matplotlib's charting tools under the short name `plt`.
- `plt.subplots(figsize=(10, 5))` creates the Figure (10 inches wide, 5 tall) and one Axes, named `fig` and `ax` (section 7.1).

READ THE OUTPUT

```
Saved: data/revenue_trend.png
```

- The Terminal only confirms the file; the chart itself is in `data/revenue_trend.png`.

Quick Check 1: `ax.grid()`

NEW TOOL

`ax.grid()`

`ax.grid(True, linestyle=":")` adds light dotted guide lines behind the chart. Why: readers can line up a point with its value on the axis.

YOUR TASK

Fill in the Blank

Add one line at the end of part **2. Process**: `ax.____(True, linestyle=":")`

Check: open `data/revenue_trend.png`; dotted grid lines now sit behind the revenue line.

Bar Chart: Marketing Spend by Month

YourName_S22_demo_2.py — Input → Process → Output

```
# 1. Input: load the monthly results
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("data/session_02_02_dataset.csv")

# 2. Process: one bar per month
fig, ax = plt.subplots(figsize=(10, 4))
sns.barplot(data=df, x="Month_Name", y="Paid_Marketing_Spend_USD_M",
            color="#475569", ax=ax)
ax.set_title("Marketing Spend Peaks in November and December")
ax.set_xlabel("")
ax.set_ylabel("Paid Marketing Spend (USD M)")

# 3. Output: save the chart
fig.savefig("data/marketing_spend_bar.png", dpi=150, bbox_inches="tight")
print("Saved: data/marketing_spend_bar.png")
```

HOW IT WORKS

- `import seaborn as sns` loads Seaborn, which draws common statistical charts with less code than Matplotlib alone.
- `sns.barplot(data=df, x="Month_Name", y="Paid_Marketing_Spend_USD_M", ...)` reads the columns by name from the DataFrame and draws one bar per month.
- Seaborn's bars start at zero, so their lengths can be compared honestly (section 7.6).

READ THE OUTPUT

```
Saved: data/marketing_spend_bar.png
```

- In the image, the bars grow through the year, and the last two are clearly the tallest: Nov `5.2` and Dec `6.4`, against `2.1` in Jan.

Quick Check 2: `.nlargest()`

NEW TOOL

`.nlargest()`

`df.nlargest(3, "Column_Name")` returns the 3 rows with the largest values in that column. Why: confirms the chart's message with exact numbers.

YOUR TASK

Fill in the Blank

Add one line to part **3. Output**: `print(df._____(3, "Paid_Marketing_Spend_USD_M")[["Month_Name", "Paid_Marketing_Spend_USD_M"]])`

Check: the Terminal lists `Dec` (`6.4`), `Nov` (`5.2`), and `Oct` (`4.1`).

Correlation Heatmap

YourName_S22_demo_3.py — Input → Process → Output

```
# 1. Input: load the monthly results
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("data/session_02_02_dataset.csv")

# 2. Process: correlation between every pair of 4 columns
cols = ["Organic_Revenue_USD_M", "Paid_Marketing_Spend_USD_M",
        "Customer_Acquisition_Cost_USD", "Net_Promoter_Score"]
corr = df[cols].corr().round(2)
fig, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap="RdBu_r", vmin=-1, vmax=1, ax=ax)
ax.set_title("All Four Measures Rose Together in 2024")

# 3. Output: print the numbers, then save the chart
print(corr)
fig.savefig("data/kpi_correlation_heatmap.png", dpi=150,
            bbox_inches="tight")
```

HOW IT WORKS

- `cols = [...]` lists the 4 measures to compare.
- `df[cols].corr()` computes the correlation of every pair, a 4 x 4 table (section 7.3).
- `sns.heatmap(corr, annot=True, cmap="RdBu_r", vmin=-1, vmax=1, ax=ax)` colors each cell: red for positive, blue for negative, white near zero.

READ THE OUTPUT

- The `...` hides middle columns that do not fit the Terminal width; the image shows all 16 cells.

Quick Check 3: `.corr()` on two columns

NEW TOOL

`.corr()` on two columns

`column_a.corr(column_b)` returns one correlation number for two columns. Why: answers a single question without a whole heatmap.

YOUR TASK

Fill in the Blank

Add one line to part **3. Output**:

```
print(round(df["Paid_Marketing_Spend_USD_M"].____(df["Organic_Revenue_USD_M"]), 3))
```

Check: the Terminal shows `0.996`.

Where to Look Things Up

REFERENCE

Matplotlib Official Documentation: User Guide

https://matplotlib.org/stable/users/index.html

REFERENCE

Seaborn Official Documentation: Statistical Data Visualization

https://seaborn.pydata.org/

REFERENCE

Edward Tufte: The Visual Display of Quantitative Information (Principles of Information Design)

https://www.edwardtufte.com/tufte/books_vdqi

Three Exercises

EXERCISE 1: GUIDED

Two-Panel Executive Dashboard

Goal: Draw two charts side by side in one image

File:
`YourName_S22_dashboard.py`

Check: the scatter climbs: correlation `0.98`

EXERCISE 2: APPLIED

Four-Panel Boardroom Dashboard

Goal: Build a 2x2 boardroom dashboard with 4 charts

File:
`YourName_S22_practice.py`

Check: 4 charts, no overlapping titles

EXERCISE 3: ADVANCED

Is Marketing Getting Less Efficient?

Goal: Does each marketing dollar still bring in as much revenue?

File:
`YourName_S22_challenge.py`

Check: average revenue per marketing dollar: `5.08`

Full steps for all three exercises: sections 11-13 of the course page.