ISA 444: Business Forecasting

03: Plotting a Single Time Series in Python

Fadel M. Megahed, PhD

Professor
Farmer School of Business
Miami University

@FadelMegahed
fmegahed
fmegahed@miamioh.edu
Automated Scheduler for Office Hours

Spring 2025

Quick Refresher of Last Class

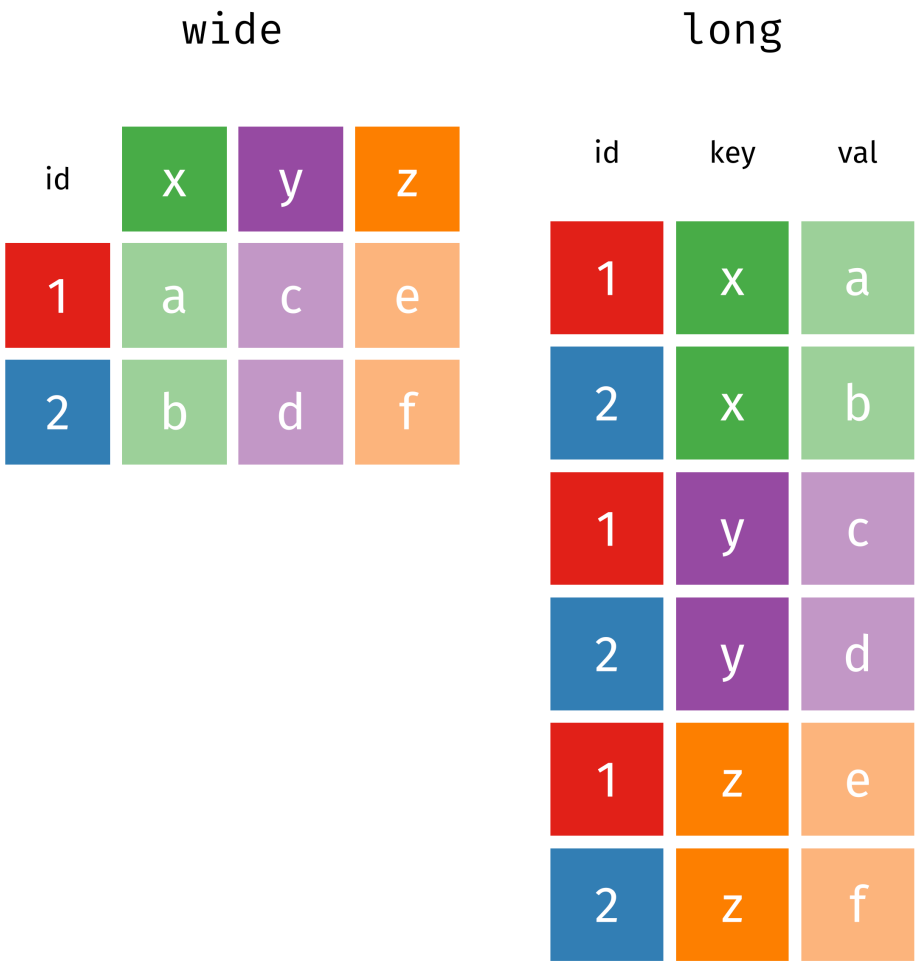
- ✅ Setting up Python (Colab, Anaconda, and/or VS Code)
- ✅ Practice basic data reading (from CSVs and the Web)
- ✅ Use panda's datatime, indexing, and slicing capabilities
- ✅ (Optional) Discuss generative AI usage in Google Colab

Learning Objectives for Today's Class

- Generate and interpret simple line charts.
- Create seasonal plots and subplots.

Generate and Interpret Simple Line Charts

Basics: Tabular Data Formats - Wide vs. Long



Source: Image is from Garrick Aden-Buie's excellent tidyexplain GitHub Repository

Basics: Tabular Data Formats - Wide vs. Long

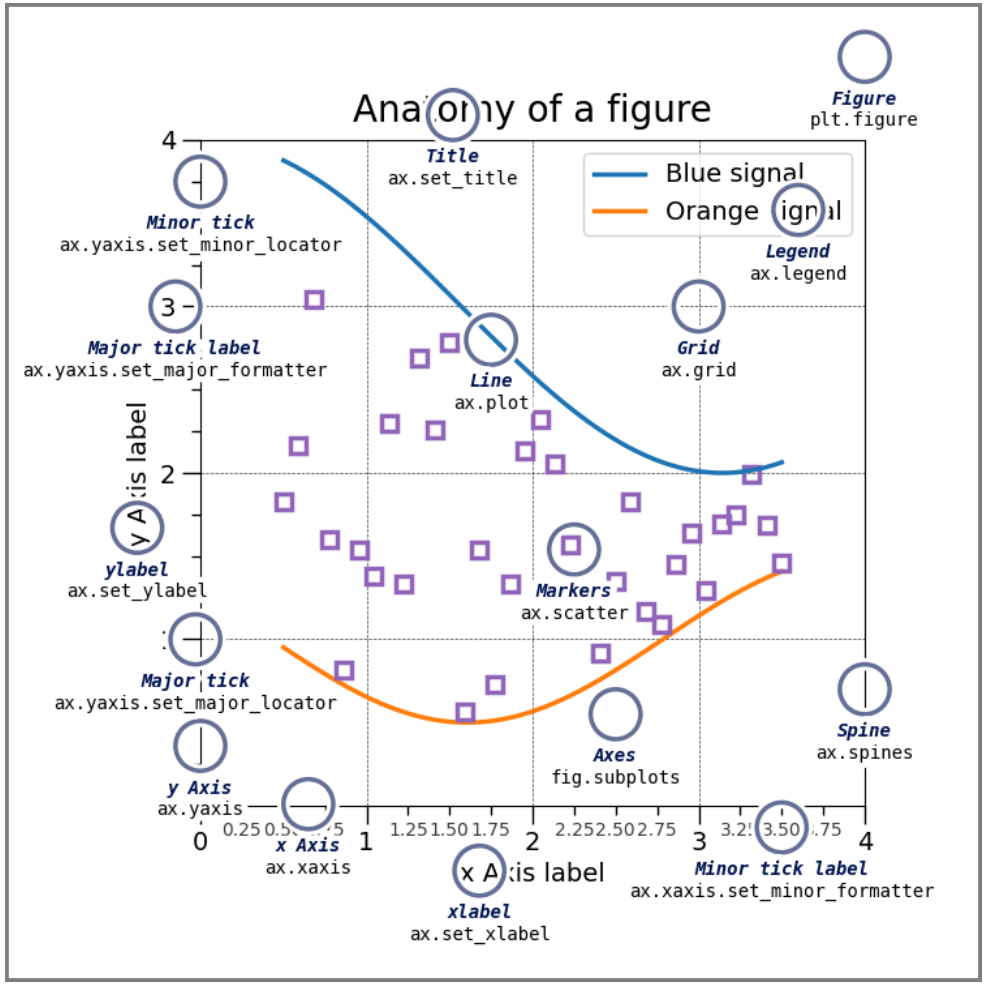
Wide Format

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

Wide Format

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Basics: The Anatomy of a Chart



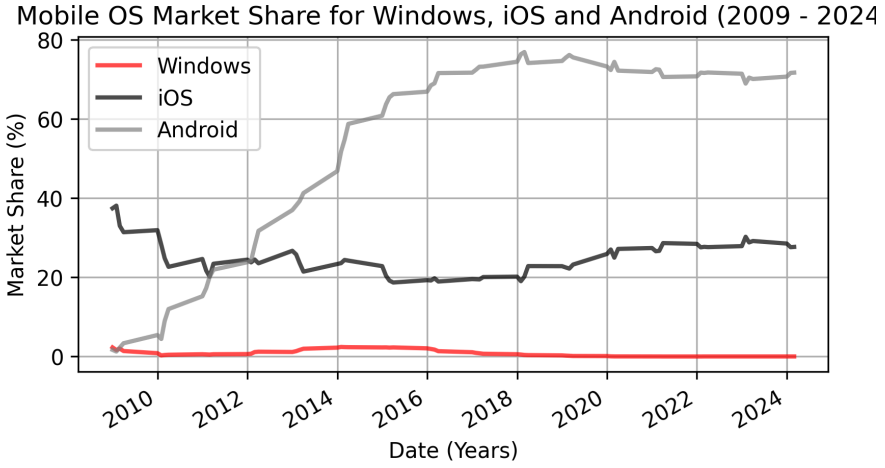
Source: Image is from the Matplotlib Documentation

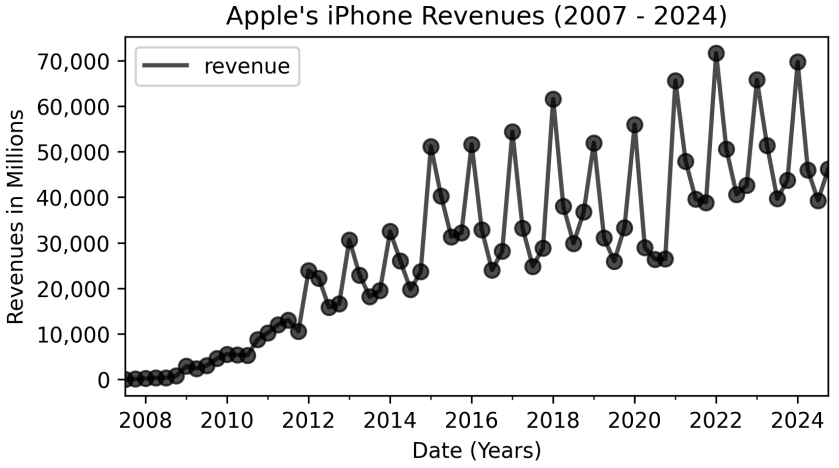
Making Charts in Python: Three Possible Approaches

- **Matplotlib**: The most basic and flexible plotting library in Python.
- **Panadas**: A wrapper around Matplotlib that makes it easier to create simple plots directly from DataFrames.
- **Seaborn**: A high-level interface to Matplotlib that makes it easier to create attractive and informative statistical graphics.

Recall: Microsoft's Missed Opportunity in Mobile Phones

- Since Q1 2009, **Windows' Mobile OS's market share <= 2.5%**, and is now at 0.02% (StatCounter).
- **Apple's Mobile iOS market share > 19%**, and is now at **27.69%** (StatCounter).
- Apple's **iPhone revenues** from 2007 to 2024 was **$2.037 trillion** (per statista).
- Assuming Microsoft could have captured just **5%** of Apple's market revenue → **$102 billion**.
- This estimate **excludes app store and brand value**, which will make the missed opportunity even larger.





So, Let's Explore Apple's IPhone Revenues

05:00

Description | Questions | Class Results | Python Code

- **Dataset**: Load Apple's iPhone revenue data from 2007 to 2024. Click here to download the data.
- Use pandas to read the data into a DataFrame.
- Answer the questions in the following tab.

Plotting the Data using Pandas

pandas | Getting started | User Guide | API reference | Development | Release notes

Search Ctrl + K | 2.2 (stable)

Input/output
General functions
Series
DataFrame
pandas.DataFrame
pandas.DataFrame.index
pandas.DataFrame.columns
pandas.DataFrame.dtypes
pandas.DataFrame.info
pandas.DataFrame.select_dt

API reference > DataFrame > pandas.DataF...

pandas.DataFrame.plot

`DataFrame.plot(*args, **kwargs)` [source]

Make plots of Series or DataFrame.

Uses the backend specified by the option `plotting.backend`. By default, matplotlib is used.

Parameters:

data : ***Series or DataFrame***

The object for which the method is called.

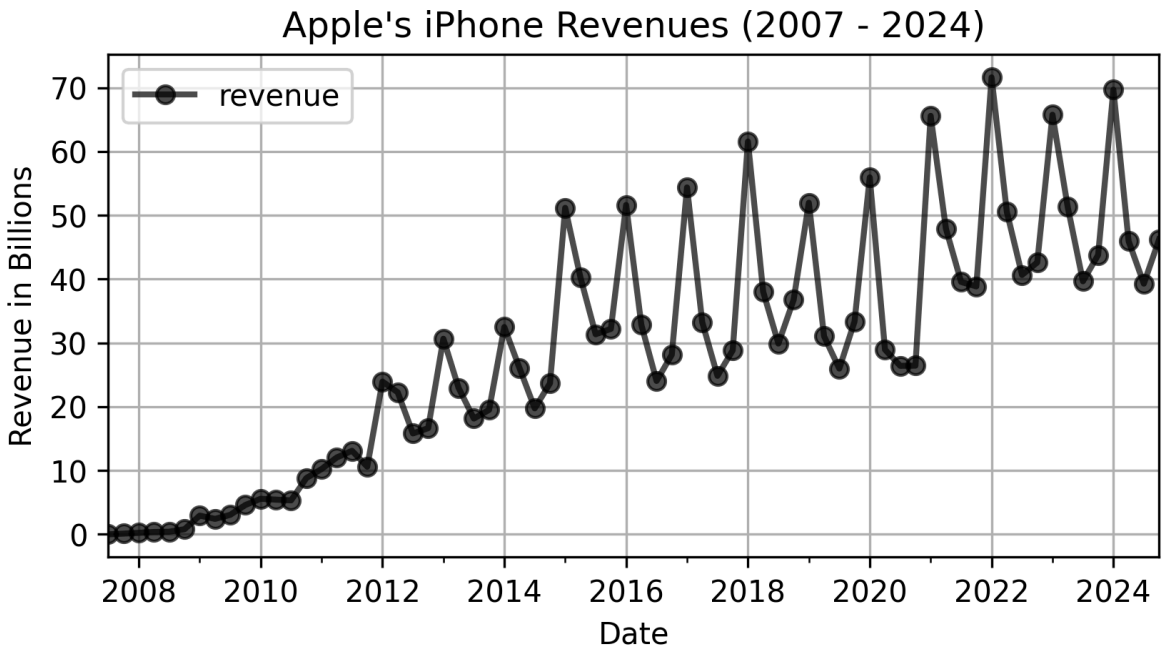
Source: Pandas API for the Plot Method

Use Pandas API to Make a Simple Line Chart

03:00

Use the Pandas Plot API to create a simple line chart of Apple's iPhone revenues from 2007 to 2024.

How close is the chart to the one you created in Python?



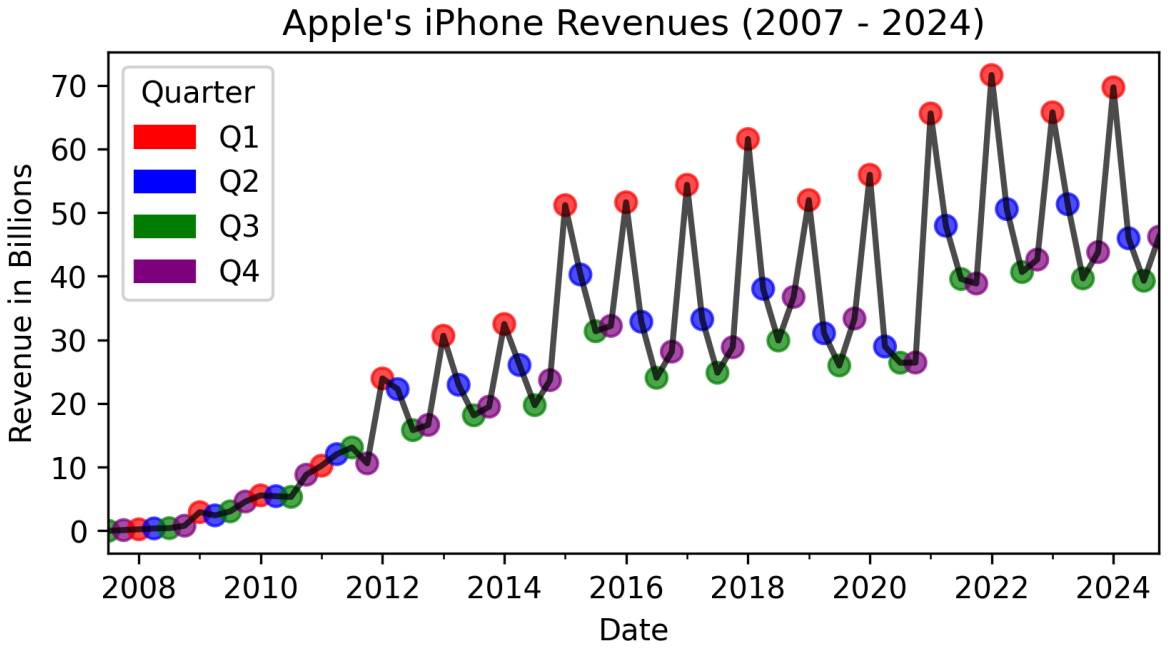
Hint: The additional **kwargs (keyword arguments) for the `plot` method can be found from `matplotlib.pyplot.plot` API Reference.

Create Seasonal Plots and Subplots

Demo: Lets Improve the Previous Plot Using Pandas

In a **live demo**, we will make the following improvements to the previous plot:

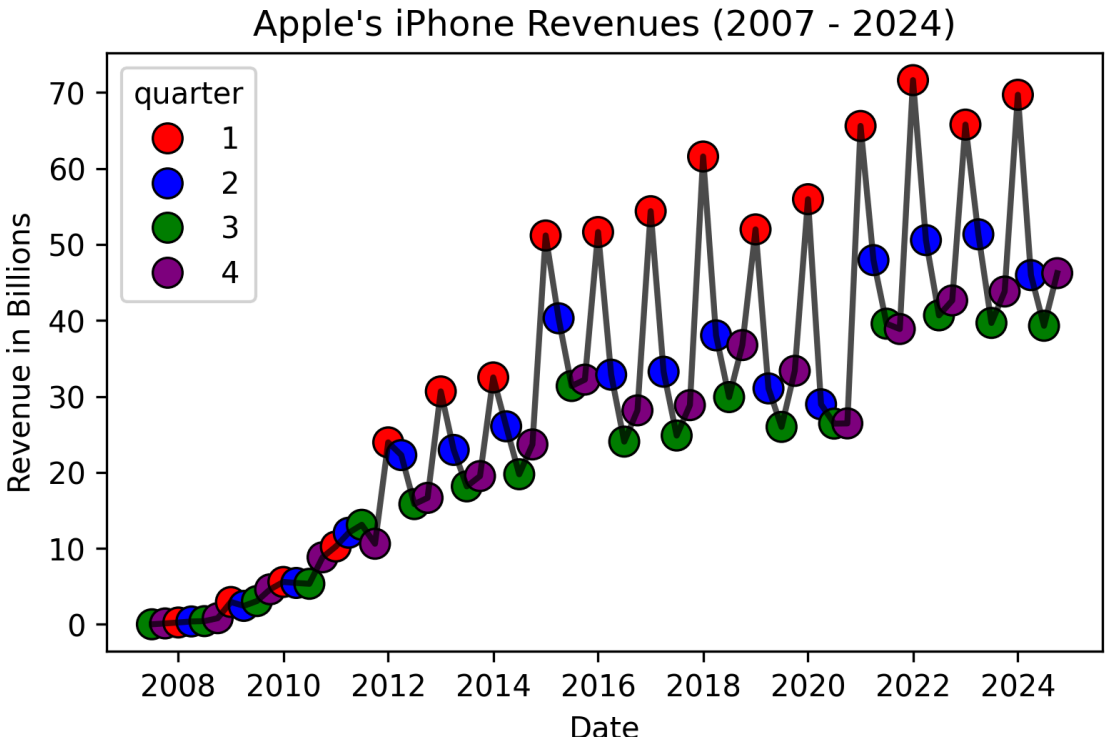
- **Color** the data points by `quarter`.
- Add a **custom legend** for the quarters.



The Equivalent Seaborn Implementation

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

apple_rev = pd.read_csv("../../data/statistic_id263402_apple-iphone-sales-revenue-2007-2024.csv")

apple_rev = apple_rev.assign(
    date=lambda df: pd.to_datetime(df['date']),
    quarter=lambda df: df['date'].dt.quarter,
    revenue=lambda df: df['revenue'].str.replace(',', '').astype(int) / 1e3
)

plt.figure(figsize=(6, 3.625)) # Create the figure

ax = sns.lineplot(data=apple_rev, x='date', y='revenue', color='black', linewidth=2, alpha=0.7)

sns.scatterplot(
    data=apple_rev, x='date', y='revenue',
    hue='quarter',    # variable to color by
    palette={1: 'red', 2: 'blue', 3: 'green', 4: 'purple'}, # custom color palette
    s=100,             # Marker size; adjust as desired
    edgecolor='black', # Optional: add a black edge around markers
)

# Customize the axes and title, and show the plot
ax.set_title("Apple's iPhone Revenues (2007 - 2024)")
ax.set_xlabel("Date")
ax.set_ylabel("Revenue in Billions")
plt.show()
```

The Equivalent Seaborn Implementation



What Observations Can You Make from the Chart?

03:00

- Edit me
- ...
- ...

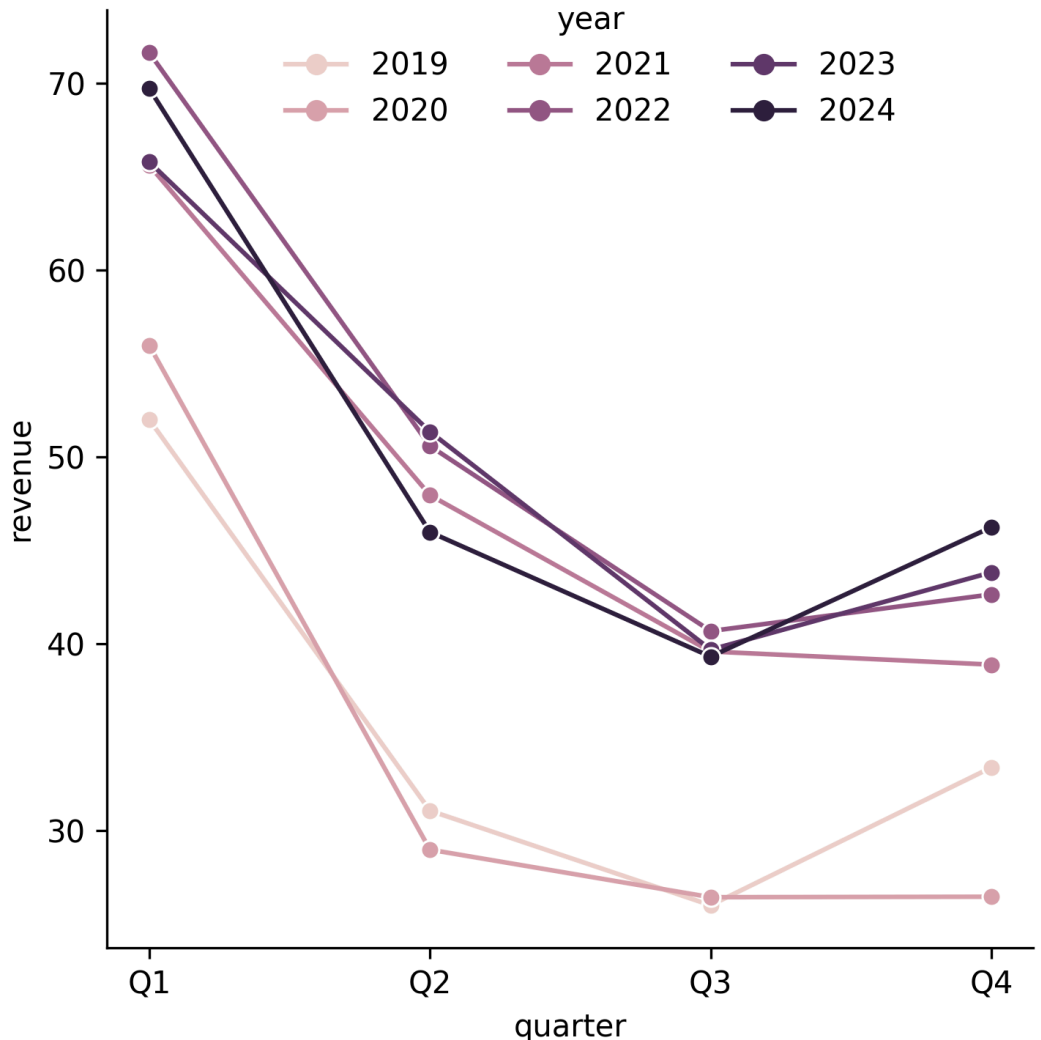
Other Types of Seasonal Plots

- **Seasonal Plot (By Month/Quarter)**: A line chart where the data is plotted against the season.
- **Seasonal Subseries Plot**: A line chart where each season is plotted separately.
- **Multiple Box Plots**: A box plot for each season.

Seasonal Plot (By Month/Quarter)

```
import pandas as pd
import matplotlib.pyplot as plt

apple_rev = (
  pd.read_csv("../../data/statistic_id263402_apple-iphone-sales-revenue-2007-2024.csv")
  .assign(
    date = lambda x: pd.to_datetime(x['date']),
    quarter = lambda x: x['date'].dt.quarter,
    year = lambda x: x['date'].dt.year,
    revenue = lambda x: x['revenue'].str.replace(r',', '').astype(int)/1e3
    )
    .query('year >= 2019')
  )

s = sns.relplot(
  kind="line",
  data=apple_rev, x="quarter", y="revenue",
  hue="year", marker="o",
  # row = "year", # optional: separate by year
)

# beautify the plot (will not work with row = 'year')
s.ax.set_xticks(range(1, 5))
s.ax.set_xticklabels(['Q1', 'Q2', 'Q3', 'Q4'])
sns.move_legend(s, "upper center", ncols = 3)

plt.show()
```

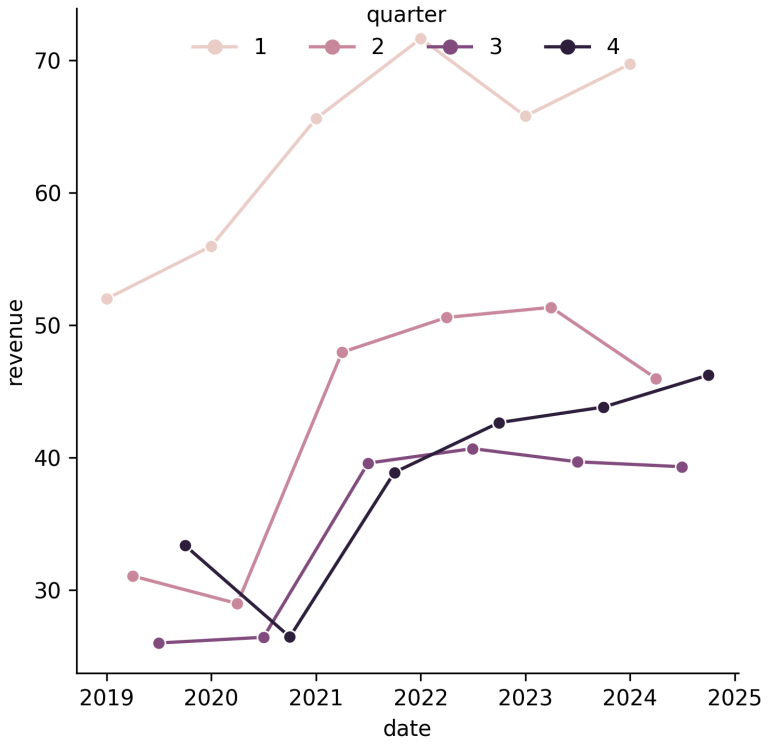
Seasonal Plot (By Month/Quarter)



Seasonal Subseries Plot

03:00

Description | Chart to Make



Recap

Summary of Main Points

By now, you should be able to do the following:

- Generate and interpret simple line charts.
- Create seasonal plots and subplots.

📝 Review and Clarification 📝

- **Class Notes**: Take some time to revisit your class notes for key insights and concepts.
- **Zoom Recording**: The recording of today's class will be made available on Canvas approximately 3-4 hours after the session ends.
- **Questions**: Please don't hesitate to ask for clarification on any topics discussed in class. It's crucial not to let questions accumulate.

📖 Required Readings 📖

- **Reference** the following pages:
  - Pandas Plot API
  - An Introduction to Seaborn
  - An Introduction to Seaborn for data viz with Python