

```
# Week 1 Lessons Program: Crypto Market Data Analysis with Pandas
**Dates**: April 26, 2025 – May 2, 2025
**Goal**: Learn to analyze and clean crypto market data (BTC, ETH, BNB) using Pandas.
**Dataset**: `crypto_market_data.csv` (fetched with `yfinance`, includes missing values and outliers).
**Daily Schedule**:
- 6:00–8:00 AM: Theory and Examples
- 12:00–2:00 PM: Exercises
- 18:00–20:00 PM: Review and Mini-Project
```

Day 1: Saturday, April 26, 2025 – Loading and Exploring Crypto Data

```
### 6:00–8:00 AM: Theory and Examples
- **Concepts**: Load a dataset into Pandas, explore its structure, compute basic statistics.
- **Code**:
  ```python
 import pandas as pd
 df = pd.read_csv("crypto_market_data.csv")
 print("Dataset shape:", df.shape)
 print("First 5 rows:\n", df.head())
 print("Column info:\n", df.info())
 print("Basic statistics:\n", df.describe())
  ```
```

12:00–2:00 PM: Exercises

1. Print the last 5 rows using `df.tail()`.
2. Print the data types of each column using `df.dtypes`.
3. Compute the mean closing price for BTC (`close_btc`) and ETH (`close_eth`).

18:00–20:00 PM: Review and Mini-Project

- **Review**: Check solutions (e.g., `close_btc` mean will be `NaN` due to missing values).
- **Mini-Project**: Create a new DataFrame with `date`, `close_btc`, `close_eth`. Print its shape and save to `btc_eth_closing.csv`.

Day 2: Sunday, April 27, 2025 – Selecting and Filtering Crypto Data

```
### 6:00–8:00 AM: Theory and Examples
- **Concepts**: Select columns/rows, filter data with conditions.
- **Code**:
  ```python
 import pandas as pd
 df = pd.read_csv("crypto_market_data.csv")
 # Select closing prices
 closing_prices = df[["close_btc", "close_eth", "close_bnb"]]
 print("Closing prices:\n", closing_prices.head())
 # Select rows with iloc
 first_5_days = df.iloc[0:5]
 print("First 5 days:\n", first_5_days)
 # Filter high-volume days for BTC
 btc_volume_threshold = df["volume_btc"].quantile(0.75)
 high_volume_btc = df[df["volume_btc"] > btc_volume_threshold]
 print("High volume days for BTC:\n", high_volume_btc[["date", "volume_btc"]])
  ```
```

12:00–2:00 PM: Exercises

1. Select `volume_btc` and `volume_eth` columns, print the first 10 rows.
2. Use `loc` to select rows where `close_btc` > its median (handle `NaN` with `notna()`).
3. Filter days where `close_eth` increased from the previous day (use `diff()`).

18:00–20:00 PM: Review and Mini-Project

- **Review**: Check solutions (e.g., use `df["close_eth"].diff() > 0` for Exercise 3).
- **Mini-Project**: Create a DataFrame with days where `volume_btc` > 90th percentile, save to

```
`high_volume_btc.csv`.
```

```
---
```

Day 3: Monday, April 28, 2025 – Handling Missing Data in Crypto Prices

6:00–8:00 AM: Theory and Examples

```
- Concepts: Detect and handle missing values (`NaN`) in `close_btc` (rows 5–7).
- Code:
  ```python
 import pandas as pd
 df = pd.read_csv("crypto_market_data.csv")
 # Detect missing values
 print("Missing values:\n", df.isna().sum())
 # Fill with mean
 df_filled = df.copy()
 df_filled["close_btc"] = df_filled["close_btc"].fillna(df_filled["close_btc"].mean())
 print("BTC prices after filling:\n", df_filled["close_btc"].head(10))
 # Interpolate
 df_interpolated = df.copy()
 df_interpolated["close_btc"] = df_interpolated["close_btc"].interpolate()
 print("BTC prices after interpolation:\n", df_interpolated["close_btc"].head(10))
  ```
```

12:00–2:00 PM: Exercises

1. Count missing values in `close_btc` and `close_bnb`.
2. Fill missing `close_btc` values with the median.
3. Compare the mean of `close_btc` before and after filling (using `df_filled`).

18:00–20:00 PM: Review and Mini-Project

```
- Review: Check solutions (e.g., 3 missing values in `close_btc`).
- Mini-Project: Create a DataFrame with no missing `close_btc` (use interpolation), compute average `close_btc` for first vs. second half of the dataset.
```

```
---
```

Day 4: Tuesday, April 29, 2025 – Cleaning Outliers in Crypto Data

6:00–8:00 AM: Theory and Examples

```
- Concepts: Detect and handle outliers in `volume_eth` (1e12 at row 10) using IQR method.
- Code:
  ```python
 import pandas as pd
 df = pd.read_csv("crypto_market_data.csv")
 # Detect outliers in volume_eth
 Q1 = df["volume_eth"].quantile(0.25)
 Q3 = df["volume_eth"].quantile(0.75)
 IQR = Q3 - Q1
 lower_bound = Q1 - 1.5 * IQR
 upper_bound = Q3 + 1.5 * IQR
 outliers = (df["volume_eth"] < lower_bound) | (df["volume_eth"] > upper_bound)
 print("Outliers in volume_eth:\n", df[outliers][["date", "volume_eth"]])
 # Replace with median
 median_volume = df["volume_eth"].median()
 df_cleaned = df.copy()
 df_cleaned.loc[outliers, "volume_eth"] = median_volume
 print("Volume ETH after replacing:\n", df_cleaned["volume_eth"].describe())
  ```
```

12:00–2:00 PM: Exercises

1. Apply IQR method to detect outliers in `close_bnb` (0 at row 15).
2. Replace outliers in `close_bnb` with the mean.
3. Compare standard deviation of `volume_eth` before and after outlier removal.

18:00–20:00 PM: Review and Mini-Project

```
- Review: Check solutions (e.g., `close_bnb` outlier at 0 should be detected).
```

- **Mini-Project**: Create a DataFrame with no outliers in `volume_eth` and `close_bnb`, save to `crypto_cleaned.csv`.

Day 5: Wednesday, April 30, 2025 - Analyzing Price Trends

6:00-8:00 AM: Theory and Examples

- **Concepts**: Compute price changes, percentage changes, identify significant movements.

- **Code**:

```
```python
import pandas as pd
df = pd.read_csv("crypto_market_data.csv")
Daily price change for BTC
df["btc_price_change"] = df["close_btc"].diff()
print("BTC price changes:\n", df[["date", "btc_price_change"]].head())
Percentage change
df["btc_pct_change"] = df["close_btc"].pct_change() * 100
print("BTC % changes:\n", df[["date", "btc_pct_change"]].head())
Significant movements (> 5%)
significant_moves = df[abs(df["btc_pct_change"]) > 5]
print("Significant BTC movements:\n", significant_moves[["date", "btc_pct_change"]])
```
```

12:00-2:00 PM: Exercises

1. Compute daily price change for ETH (`close_eth`).
2. Calculate percentage change for BNB (`close_bnb`).
3. Find days where ETH price dropped by more than 3%.

18:00-20:00 PM: Review and Mini-Project

- **Review**: Check solutions (e.g., use `df["close_eth"].pct_change() * 100 < -3` for Exercise 3).

- **Mini-Project**: Create a summary table with average percentage change for BTC, ETH, BNB, save to `price_changes.csv`.

Day 6: Thursday, May 1, 2025 - Working with Time-Series Crypto Data

6:00-8:00 AM: Theory and Examples

- **Concepts**: Convert to datetime index, resample data, compute rolling means.

- **Code**:

```
```python
import pandas as pd
df = pd.read_csv("crypto_market_data.csv")
Convert to datetime index
df["date"] = pd.to_datetime(df["date"])
df.set_index("date", inplace=True)
Resample to weekly averages
weekly_avg = df[["close_btc", "close_eth", "close_bnb"]].resample("W").mean()
print("Weekly averages:\n", weekly_avg)
3-day rolling mean for BTC
rolling_avg = df["close_btc"].rolling(window=3).mean()
print("3-day rolling avg for BTC:\n", rolling_avg.head(10))
```
```

12:00-2:00 PM: Exercises

1. Resample to weekly sums for `volume_btc`.
2. Compute a 5-day rolling mean for `close_eth`.
3. Filter data for the last week (April 24-April 30, 2025).

18:00-20:00 PM: Review and Mini-Project

- **Review**: Check solutions (e.g., use `df["2025-04-24":"2025-04-30"]` for Exercise 3).

- **Mini-Project**: Resample to weekly averages for all closing prices, save to `weekly_crypto.csv`.

Day 7: Friday, May 2, 2025 – Combining Crypto Data with Sentiment Data

6:00–8:00 AM: Theory and Examples

- **Concepts**: Create a sentiment dataset, merge with crypto data.

- **Code**:

```
```python
import pandas as pd
import numpy as np
df = pd.read_csv("crypto_market_data.csv")
df["date"] = pd.to_datetime(df["date"])
Create sentiment dataset
sentiment_data = {
 "date": pd.date_range(start="2025-03-28", end="2025-04-26", freq="D"),
 "btc_sentiment": np.random.uniform(-1, 1, 30)
}
sentiment_df = pd.DataFrame(sentiment_data)
Merge DataFrames
merged_df = pd.merge(df, sentiment_df, on="date", how="inner")
print("Merged DataFrame:\n", merged_df[["date", "close_btc", "btc_sentiment"]].head())
```
```

12:00–2:00 PM: Exercises

1. Create a DataFrame with 5 rows containing `date` and `eth_sentiment`.
2. Merge with the crypto DataFrame using an inner join.
3. Compute correlation between `close_btc` and `btc_sentiment` (use `corr()`).

18:00–20:00 PM: Review and Mini-Project

- **Review**: Check solutions (e.g., use `merged_df[["close_btc", "btc_sentiment"]].corr()` for Exercise 3).
- **Mini-Project**: Analyze average `close_btc` for positive vs. negative sentiment days, save to `sentiment_analysis.csv`.

Summary

- **Topics Covered**: Loading/exploring data, selecting/filtering, handling missing values, cleaning outliers, analyzing trends, time-series, combining datasets.
- **Next Steps**: Complete Week 1 by May 2, 2025. Request Week 2 program for visualization or deeper analysis.