

Module 1 Summary: AI Programming Foundations

Tyler Wilcoxson

Udacity

AI Mastery Capstone

February 2026

Module 1 Summary: AI Programming Foundations

Overview

This project implements a reproducible data workflow using Python, Pandas, and Seaborn to analyze the NYC Airbnb Open Data (2019) dataset. The workflow covers the full data pipeline: ingestion, cleaning, exploratory data analysis, and visualization. The goal is to demonstrate foundational programming and data analysis skills that underpin future machine learning, deep learning, and generative AI work.

The project is implemented in a Jupyter notebook (`data_workflow.ipynb`) with all transformations encapsulated in documented Python functions. Version control is managed with Git, and all dependencies are pinned in a `requirements.txt` file to ensure reproducibility.

Dataset Description

The NYC Airbnb Open Data dataset contains 48,895 rows representing individual short-term rental listings across New York City's five boroughs. Each row includes 16 columns: listing identifiers (`id`, `name`, `host_id`, `host_name`), location data (`neighbourhood_group`, `neighbourhood`, `latitude`, `longitude`), listing characteristics (`room_type`, `price`, `minimum_nights`, `availability_365`), and review metrics (`number_of_reviews`, `last_review`, `reviews_per_month`, `calculated_host_listings_count`).

Key data quality issues include approximately 10,052 missing values in `reviews_per_month` and `last_review` (corresponding to listings that have never been reviewed), minor null values in `name` and `host_name` fields, and price outliers ranging from \$0 to \$10,000 per night.

Workflow Description

The workflow follows five sequential stages. First, data ingestion loads the CSV file and produces an initial assessment of shape, data types, missing values, and summary statistics. Second, data cleaning addresses missing values and removes price outliers through two documented functions: `handle_missing_values()` fills `reviews_per_month` with zero, parses `last_review` to datetime, and fills null `name`/`host_name` fields; `remove_price_outliers()` filters out \$0 listings and those above the 99th percentile.

Third, exploratory data analysis uses a reusable `summarize_by_group()` function to compute grouped statistics (count, mean, median, standard deviation, min, max) across borough and room type dimensions. A cross-tabulation and correlation matrix provide additional analytical perspectives. Fourth, three visualizations present key findings: a box plot of price distributions by borough, a stacked bar chart of room type composition, and a geographic scatter plot colored by price. Fifth, a summary section consolidates key insights, patterns, assumptions, and limitations.

Key Decisions and Assumptions

Data cleaning decisions were guided by tidy data principles, which require that each variable form a column, each observation form a row, and each type of observational unit form a table (Wickham, 2014). The `reviews_per_month` null values were filled with zero rather than dropped because these represent listings with no review activity, and zero is semantically accurate. Dropping these rows would remove over 20% of the dataset and introduce selection bias toward heavily-reviewed listings.

Price outlier removal used two thresholds: a lower bound of \$1 (removing \$0 listings as likely inactive or erroneous) and an upper bound at the 99th percentile (removing extreme prices that distort summary statistics and visualizations). This approach preserves 98% of the data while eliminating values that would compress visual scales and inflate mean calculations.

The exploratory analysis focused on borough-level and room-type comparisons because these represent the primary dimensions of variation in the NYC rental market. Visualization design choices, including the use of the `viridis` and `plasma` colormaps, prioritize perceptual uniformity and accessibility for colorblind readers.

Results and Interpretation

Three key findings emerged from the analysis. First, Manhattan dominates pricing with the highest median nightly rate and the widest interquartile range, reflecting its diverse accommodation market from budget shared rooms to luxury apartments (Figure 1). Brooklyn follows as the second most expensive borough, while Staten Island, the Bronx, and Queens display compressed price distributions with lower medians.

Second, room type composition varies significantly by borough (Figure 2). Manhattan has the highest proportion of entire home/apartment listings (approximately 60%), consistent with its role as a hotel-alternative market. Outer boroughs skew toward private rooms, suggesting hosts in these areas are more likely to rent a room in their own residence rather than dedicate an entire unit to short-term rental.

Third, the geographic scatter plot (Figure 3) reveals that high-price listings cluster in lower and midtown Manhattan and along the Brooklyn waterfront near Williamsburg and DUMBO. Price gradients radiate outward from these centers, with outer boroughs showing predominantly lower and more uniform pricing.

Responsible Practice

Several sources of bias should be acknowledged. The dataset captures only Airbnb listings, excluding other platforms and the traditional hotel market, which limits generalizability. The 2019 snapshot cannot capture temporal dynamics or post-pandemic shifts. Missing review data affects over 20% of listings, and filling with zero treats never-reviewed listings as inactive, potentially underrepresenting new or niche properties.

Prices are self-reported by hosts and may not reflect actual transaction prices, seasonal discounts, or negotiated rates. The 99th percentile outlier threshold, while common, is arbitrary and excludes legitimate ultra-luxury listings. These decisions shape the analytical frame and should be disclosed to anyone building on this work.

Reproducibility

Reproducibility is a core design goal of this project, following principles outlined in open learning resources for reproducible data science (Danchev, 2022). All data transformations are implemented in documented Python functions with docstrings describing parameters, return

values, and rationale. Dependencies are pinned to specific versions in requirements.txt (pandas 3.0.0, seaborn 0.13.2, numpy 2.4.2, matplotlib 3.10.8).

The dataset is included in the Git repository so the notebook runs immediately after cloning without requiring external downloads or API keys. The project uses a feature-branch Git workflow with multiple commits per development stage, providing a clear development history. Anyone can reproduce the full analysis by cloning the repository, installing dependencies, and running the notebook top-to-bottom.

References

Danchev, V. (2022). Reproducible data science with Python: An open learning resource. *Journal of Open Source Education*, 5(56), 156. <https://doi.org/10.21105/jose.00156>

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10), 1-23. <https://doi.org/10.18637/jss.v059.i10>