# Module 1 Summary: AI Programming Foundations

Tim Wilcoxson
Udacity
AI Mastery Capstone
February 2026

**Module 1 Summary: AI Programming Foundations**

## Overview

For this project I built a reproducible data workflow in Python to analyze the NYC Airbnb Open Data (2019) dataset. The workflow covers the full pipeline: ingestion, cleaning, exploratory data analysis, and visualization. The point was to practice the foundational programming and data analysis skills that come up again in the later machine learning, deep learning, and generative AI modules.

Everything is implemented in a Jupyter notebook (data_workflow.ipynb) with the data transformations wrapped in documented Python functions. I used Git for version control and pinned all dependencies in a requirements.txt file so anyone can reproduce the results.

## Dataset Description

The NYC Airbnb Open Data dataset has 48,895 rows, where each row is a short-term rental listing in one of New York City's five boroughs. There are 16 columns: listing identifiers (id, name, host_id, host_name), location data (neighbourhood_group, neighbourhood, latitude, longitude), listing details (room_type, price, minimum_nights, availability_365), and review metrics (number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count).

The main data quality issues I noticed were about 10,052 missing values in reviews_per_month and last_review (these are listings that have never been reviewed), a handful of null name and host_name fields, and price outliers ranging from $0 all the way up to $10,000 per night.

## Workflow Description

The workflow has five stages that run in sequence. First, I load the CSV file and take a look at the shape, data types, missing values, and summary statistics to get a sense of what I'm working with.

Second, I clean the data using two functions. handle_missing_values() fills reviews_per_month with zero, parses last_review into a proper datetime, and fills the few null name/host_name fields with "Unknown." remove_price_outliers() drops the $0 listings and anything above the 99th percentile.

Third, I run exploratory data analysis using a summarize_by_group() function that computes grouped statistics (count, mean, median, standard deviation, min, max) across borough and room type. I also built a cross-tabulation and correlation matrix to look at relationships between the numeric columns.

Fourth, I created three visualizations: 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 pulls together the key insights, patterns, assumptions, and limitations.

## Key Decisions and Assumptions

I based my cleaning decisions on tidy data principles (Wickham, 2014), where each variable gets its own column, each observation is a row, and each type of data forms its own table. For the reviews_per_month nulls, I filled them with zero rather than dropping the rows, since these are listings that genuinely have no review activity - zero is the right value. Dropping those rows would have cut over 20% of the dataset and biased things toward heavily-reviewed listings.

For price outliers, I used two thresholds: a lower bound of $1 (to remove $0 listings that are probably inactive or erroneous) and an upper bound at the 99th percentile (to remove extreme prices that would throw off the summary stats and compress the visualizations). This keeps about 98% of the data while getting rid of the values that would distort the analysis.

I focused the exploratory analysis on borough-level and room-type comparisons because those turned out to be the biggest sources of variation in the dataset. For the visualizations, I went with the viridis and plasma colormaps since they work well for colorblind readers and have consistent brightness gradients.

## Results and Interpretation

Three main findings came out of the analysis. First, Manhattan dominates pricing - it has the highest median nightly rate ($149) and the widest interquartile range, which makes sense given that it has everything from budget shared rooms to high-end apartments (Figure 1). Brooklyn is the second most expensive borough, while Staten Island, the Bronx, and Queens all have tighter price ranges and lower medians.

Second, room type composition is pretty different across boroughs (Figure 2). Manhattan has about 60% entire home/apartment listings, which makes sense since a lot of those are basically being used as hotel alternatives. The outer boroughs lean more toward private rooms, suggesting hosts there are more likely renting out a spare room rather than dedicating a whole unit to short-term rental.

Third, the geographic scatter plot (Figure 3) shows that high-price listings are concentrated in lower and midtown Manhattan and along the Brooklyn waterfront near Williamsburg and DUMBO. Prices generally drop off as you move into the outer boroughs, where the distribution is more uniform and mostly on the lower end.

## Responsible Practice

There are several places where bias could creep into this analysis. The dataset only captures Airbnb listings - it doesn't include VRBO, Booking.com, or traditional hotels - so any conclusions are limited to one platform's slice of the market. It's also a 2019 snapshot, so it can't capture anything about how the pandemic changed the rental landscape.

The missing review data affects over 20% of the listings. Filling those with zero treats never-reviewed listings the same as inactive ones, which could undercount newer listings or ones in less popular neighborhoods. Prices are also self-reported by hosts and might not line up with what guests actually paid after discounts or negotiations.

The 99th percentile outlier cutoff is common but ultimately arbitrary - picking a different threshold like the 95th or 99.5th percentile would change the summary statistics. I think these are reasonable choices for this project, but anyone building on this work should know about them.

## Reproducibility

I tried to make this project fully reproducible, following the kind of practices Danchev (2022) describes for open data science workflows. All the data transformations are wrapped in Python functions with docstrings that explain what they do and why. 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 right after cloning without needing any external downloads or API keys. I used a feature-branch workflow with multiple commits per development stage so there's a clear record of how the project evolved. Anyone should be able to reproduce the full analysis by cloning the repo, 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