

The Market Comps Playbook

This is the playbook for turning an MLS export into a market analysis your sellers have never seen from another agent. Most comps packets stop at the subject property's own street. This one pulls the surrounding cities into the picture too, so you can show a seller or a buyer something with real weight behind it: what the same house is worth one city over, and why that matters for the price you're setting or the offer you're writing.

What You're Building

Two things, together. First, the standard piece: closed comps on the subject property itself. Second, the piece almost nobody builds: a market picture across the 5 to 8 cities that surround it, actives and closeds both, broken down by price per square foot, days on market, and what a specific type of home actually sells for in each one.

Put those two together and you get a listing presentation or a buyer consult that looks different from anything else that agent has seen. You can literally show a buyer coming from a pricier neighboring city why the subject city is a value, with numbers, not opinion.

Want to see it built out before you build your own? I put together a live one-page example here: <https://joshuasun.co/aitools/examples/market-report/>. That's a neighborhood snapshot done this way, start to finish.

Step 1: Pull the Data From Your Own MLS

You need two exports, both from your own MLS, under your own license. Never scrape a portal, a third-party site, or someone else's report to build this. It has to be your data.

- **Export 1: subject city, 90-day closed sales.** Every closed sale in the subject property's city over the last 90 days.
- **Export 2: surrounding cities, actives plus closeds.** Same field set, but for the ring of cities around the subject city, and this one includes active listings as well as closed ones so you can see current inventory, not just what already sold.

Pick your 5 to 8 surrounding cities the same way a buyer actually shops, not by drawing a random circle on a map. Look for:

- Cities in the same commute-shed. If someone would cross-shop the subject city against it on a daily-drive basis, it's in.
- Cities that pull the same school-shopper. If the school district or feeder pattern draws the same type of buyer, include it.
- Cities that are price-adjacent. A city 3x the price isn't a comp set, it's a different market. Stick to neighbors close enough in price that a buyer would genuinely cross-shop.

Export both pulls as CSV. Your MLS field set will typically include: Listing ID, status, sub type, street number and name, city, area, list and sale price, price per square foot, beds/baths, square footage, year built, lot size, days on market (and cumulative days on market if it's been relisted), sale-to-list indicators, and the contract or close date. Keep that full field set in the export. You'll want most of it.

Step 2: Hand It to AI With the Right Questions

Load both CSVs into Claude (or your AI tool of choice) and ask it specific questions, not "analyze this." A vague prompt gets you a vague answer. Here's the actual list I run:

1. What's the median sale price and median price per square foot for each city in this data set?
2. What's the average and median days on market by city?
3. What's the average sale-to-list price ratio by city, meaning are homes selling over, at, or under asking?
4. For a home matching this profile (for example: 2,400 square feet, 4 bed, built after a certain year), what did comparable homes actually sell for in each of these cities?
5. What's the price band distribution in each city, meaning how many sales fall under \$X, between \$X and \$Y, and above \$Y?
6. Based on the active listing count and the recent closed pace, what's the rough months of inventory in each city?
7. Is there a meaningful price spread between renovated or newer-build homes and original-condition homes in this data?
8. Which city in this set is the best value on a price-per-square-foot basis for a buyer who wants what the subject city offers?
9. Are there any outlier sales pulling a city's median up or down, and what happens to that city's number if you exclude them?
10. Summarize the three numbers I'd want on one page if I were presenting this to a seller or a buyer today.

Two rules that matter more than the prompts themselves. First, make the AI show its work. Ask it to cite which rows or which addresses it pulled a number from, every time. Second, spot-check a handful of those numbers against the raw CSV yourself before anything goes in front of a client. If it says the median in a city is a certain number, open the file and confirm a few of the underlying sales. This step takes two minutes and it's the difference between a tool and a liability.

Step 3: The Nuances That Separate a Real Analysis From a Chart Dump

This is the part that actually matters. Anybody can ask AI to average a column. What makes this useful is knowing where the numbers lie to you if you're not careful.

- **Actives are not values.** An active listing is an ask, not a sale. When you blend actives and closeds, always label which is which. Never let "average list price" quietly become "average value" in your presentation.
- **Same-basis rule.** Never mix a 90-day median with a 12-month median, or one city's MLS pull with another source's, and call it one number. If the date ranges or the sources don't match, say so, or don't compare them.
- **Small-sample humility.** If a city only produced 6 or 7 closed sales in your window, that's a story, not a statistic. Say so in the presentation instead of presenting it with the same confidence as a city with 80 sales.
- **Condition and renovation deltas.** A city's median can look soft simply because more of its recent sales were original-condition homes. Separate renovated from original before

you draw a conclusion about a city being "cheaper."

- **Lot size and ADU potential.** Two homes with the same square footage on very different lot sizes are not the same comp, especially in markets where ADU potential adds real value. Note lot size, don't just compare structure square footage.
- **Watch for the luxury pocket dragging a median.** One \$3M sale in a city that's mostly \$700K homes will pull the average hard. Always look at the median next to the average, and ask the AI to flag outliers before you trust either number.
- **Stamp everything.** Every number that leaves your laptop needs an as-of date and a source. "As of [date], based on [X]-day MLS export." No exceptions.

Step 4: What to Do With It

Listing presentations. Price the home using the subject city's own comps, then back it up with the neighboring-cities picture. Showing a seller that the identical home is worth meaningfully more one city over is a different conversation than showing them three comps on their own street.

Buyer consults. Use the same data to show a buyer the value city. If they're comparing the subject city against a pricier neighbor, the price-per-square-foot comparison does more convincing than anything you could say.

Farming content. Turn this into a monthly one-pager for your farm area. Same method, same questions, refreshed each month. If you want it to look like a branded piece instead of a spreadsheet, ask your AI tool to design it as a one-pager, the same way the example above is built.

The Lanes

Keep this inside a few clear lines and you're fine.

- Every number comes from your own MLS pull under your own license. Never republish another agent's data or a scraped portal number as your own analysis.
- Never publish a value estimate on someone else's active listing. This is market analysis for your own clients, not a public opinion of what another agent's listing is worth.
- This is an analysis of properties and prices, never of people. If a client ever asks something like "what kind of people live there," the answer is data about homes, schools as institutions, and prices. It is never demographics. Steering a client using who lives somewhere is a fair housing violation, full stop.

Want the Analysis Without the Homework?

Text me at **858-216-6091** and I'll run one with you.

Joshua Sun · joshuasun.co/aitools