Beautiful data: The housing crisis in CA

Deborah F. Swayne and Hadley Wickham

December 16, 2008

Introduction

There has been much talk of the housing crisis in the media, and much speculation who has been worst hit and how long it might last. In this chapter we're going to take a more quantitative look at the housing crisis by exploring data about the sales half a million homes in the Bay Area. We are going to use a few simple statistical tools, but mostly we will focus on graphical displays of the data. (You might wonder why we chose to explore the Bay Area given that none of us live there, but it revolves around the availability of the data.)

This is a large data and we have only just scratched the surface. If the data has caught your interest and you'd like to follow our work in more detail and try out some your own ideas, we have made all data and code available in a git repository at `https://github.com/hadley/sfhousing`. Both code and data are licensed with the permissive MIT license. We'll illustrate snippets of code and fragments of the data inline, but each section of this chapter will also point you to a matching file online. The files don't cover the exactly content as each section: you can see some of the dead ends we tried, and some of the plots that weren't quite good enough to make it into the plot.

We're going use R, a statistical programming and data analysis environment, for most of the analysis. We'll discuss R and a few other we used in more depth in the conclusion, but for now lets dive into the data.

How did we get the data?

Finding relevant datasets for a particular problem is challenging and requires a lot of exploration and investigation. We were particularly luckily to stumble over the the weekly housing sales update for the Bay Area produced by the SF Chronicle, `http://www.sfgate.com/homesales/`. Initially we had planned on scraping the data off the website, but a little detective work revealed that the data is already available in a machine readable form. Each human readable weekly summary points to a machine readable text file that looks like this:

```
rowid: 1
county: Alameda County
city: Alameda
newcity: 1
zip: 94501
street: 1220 Broadway
price: $509,000
br: 4
lsqft: 4420
bsqft: 1834
year: 1910
```

Each weeks worth of data is available at a url of the form `http://www.sfgate.com/c/a/year/month/day/REHS.tbl`. This is pretty convenient, but we need to generate a list of all Sundays from the first on record, 2003-04-27 (which we found on the archive page), to the most recent, 2008-11-16. We then generated a list of urls in the correct format and then downloaded them with command line tool `wget`. We used wget because if interrupted it can easily resume where it left off: this saves a lot of time!

With all the data on a local computer, the next step was to convert the data into a more standard format. We use csv (comma separated values) for most datasets: although there's no standards format that exactly describes the structure of csv files, for most statistical data sets it's not complicated and every statistical package (and Excel!) can read it in without problems. This gives us a file like follows:

```
county,city,zip,street,price,br,lsqft,bsqft,year,date,datesold
Alameda County,Alameda,94501,1220 Broadway,509000,4,4420,1834,1910,2003-04-27,NA
Alameda County,Alameda,94501,429 Fair Haven Road,504000,4,6300,1411,1964,2003-04
-27,NA
Alameda County,Alameda,94501,2804 Fernside Boulevard,526000,2,4000,1272,1941,200
3-04-27,NA
Alameda County,Alameda,94501,1316 Grove Street,637000,3,2700,1168,1910,2003-04-2
7,NA
```

This is a little less human readable (there's much less white space), but it's a standard format that we can easily work with. Another minor advantage compared to the original format was reduced storage requirements: a 50% reduction from 90 to 45 meg. If you look at the sample data you might notice something that needs a little explanation: the **NA**s. These are what R uses to represent missing values.

It takes just a few minutes to parse all 293 data files and get `house-sales.csv`, a csv file with 521,726 observations and 11 variables. However, it tooks a few hours of tweaking the parser to get all the edge cases right: we needed to convert prices to regular numbers (by out $ and ,), parse the dates into a consistent format, and fill in missing values for fields that didn't occur in all of the tables.

We wrote a series of R and shell scripts to perform all these tasks. This is a lot of work but really pays off when your data is changing. In our case, we updated just before writing up the paper so that we had the latest data off the website. Data changes a lot more than you might thing. Even when your data is about something that has already happened, often the data will change as errors are discovered and fixed.

Geocoding

When we first looked at this data, we thought it would be really important to geocode all the addresses. That is, we wanted to associate a latitude and longitude with each address so that it would be easy to explore fine grained spatial effects. In the end, we didn't end up using this extra data as much as we thought we would, but it's still an interesting challenge: how can you geocode approximately 400,000 addresses?

We started by looking at the well known web services provided by google and yahoo. These were no good for two reasons: strict daily limits on number of requests and heavy restrictions on what we could do with the data. The daily limits meant that it would take well over a month to geocode all the addresses, and then the licensing would mean that we couldn't publish our results! After a lot of googling, we found a much more open service, the USC WebGIS Services provided by the GIS research laboratory at the University of Southern California, `https://webgis.usc.edu/`. This service is free for non-commercial use and makes no restrictions on the uses to which you can put the data. It has no daily usage cap, but there is an implicit cap caused by the speed: we could only geocode XXXX addresses per day, so it took us XX days to do all 400,000.

As well as latitude and longitude, the results also include an indication of how accurate the geocoding is. 10% percent of the addresses were located exactly based on city records for that street number (extremely accurate), another 75% percent were located by interpolating between the numbers at each end of a city

block (very accurate), 7% to the centre of the zip code (not very accurate) and the remainder were only located to the centre of the city or not at all.

It's worth spending a lot of time with this data to ensure it's accurate. If it's not, any problems will propagate through to the rest of our analysis. We discovered quite a few unusual locations! However, we will omit most of the work we did because it's more interesting to talk about the findings. Regardless, we never want to completely throw out bad matches, because we need varying levels of accuracy for different purposes: city level accuracy is fine when we are comparing cities, will want address level when we are looking within a city, or focusing on purely geographical comparisons. Instead of removing the record, we use R's missing values to indicate that we don't really know the precise location of that address. This ensures that any location with a suspicious geocoding will automatically be dropped from any analysis that uses latitude and longitude.

Analysis

We'll start by looking at number of sales and average prices over all of California, and then zoom into focus on regions of interest. Finally, we'll look at a few plots that reveal features of the data that allow us to investigate other housing patterns in California.

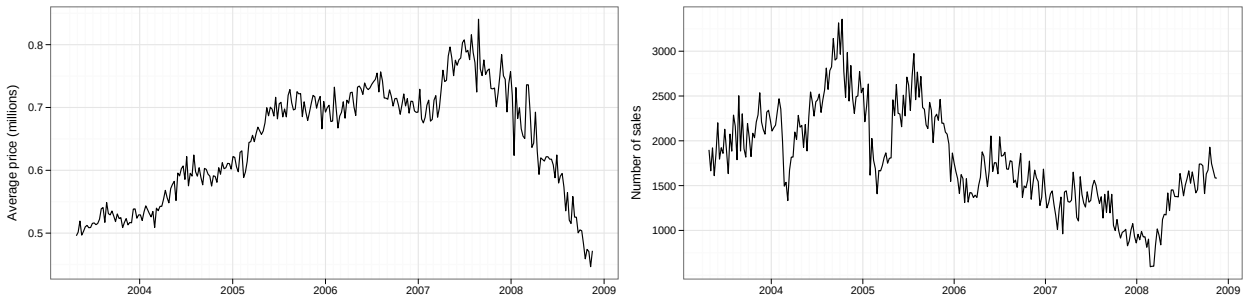

Figure 1: Daily average prices (right) and sales (left)

Figure 1 shows average daily sales and prices over the course of the data. The effect of the housing crisis is striking in the average sale prices, with an increasing trend until June 2007 and then a rapid decrease. Sales show a different pattern. We see a gradual decrease in number of sales from mid 2006, and then an increasing trend in early 2008. Maybe this is because of people buying foreclosed houses? We'll look at that in more detail later on.

Adjusting for inflation

CPI data - `http://www.bls.gov/CPI`. This is a common theme in data analysis: we need to combine our original data with new data that provides context and helps us understand it better. In this case, we want to control for changes in relative buying power (i.e. inflation). Even over this relatively short time span it's a bad idea to ignore inflation: $1 000 2004 dollars would be worth$1 170 today.

We used the CPI time series for the West coast. Series name.

To compute a measure of inflation, we index this time series at the first date in the data. Indexing means we divide all of the values by the first value: this converts the values into proportions, which makes it easy to read the effect of inflation from the graph. A value of 1.1 represent a cumulative inflation of 10% from the start of the data. Indexing is a very useful technique and we'll use it throughout this chapter.

Figure 2 shows the CPI-based inflation measurement and the affected of adjusting for inflation. Failing to adjust for inflation makes the increasing trend prior to mid 2007 look more pronounced. All prices in the

remainder of this paper will be inflation adjusted.

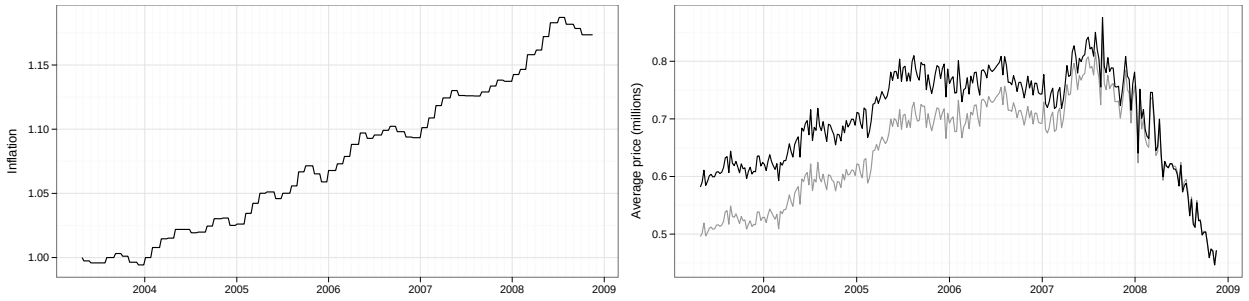

Figure 2: (Left) Inflation, indexed at 1 at start of series. (Right) Inflation adjusted prices (black), and unadjusted prices (grey). Failing to adjusting for inflation makes the rise look steeper, but has little effect on the decline.

The rich get richer and the poor get poorer

We were also interested to investigate whether the housing crisis has equally affect the rich (people owning the most expensive houses) and the poor (the people only the least expensive houses). To explore this, for each month of sales, we calculate price deciles. The deciles are the nine numbers that split the price into ten bins each containing the same number of houses, from least expensive to most expensive. For example, ...

Figure 3 shows how the deciles have changed over time. The top line is the number that 90% of houses are less than, and the bottom line represents the price that 10% of houses are cheaper than. We can see the effect of the housing bubble in mid 2007, particularly in the most expensive houses. However, it's probably better to look at the indexed values, as in Figure **??**. Here we see that the cheaper houses peaked earlier and had a much sharper drop. The cheapest houses lost 43% of their value compared to only 9% for the most expensive houses.

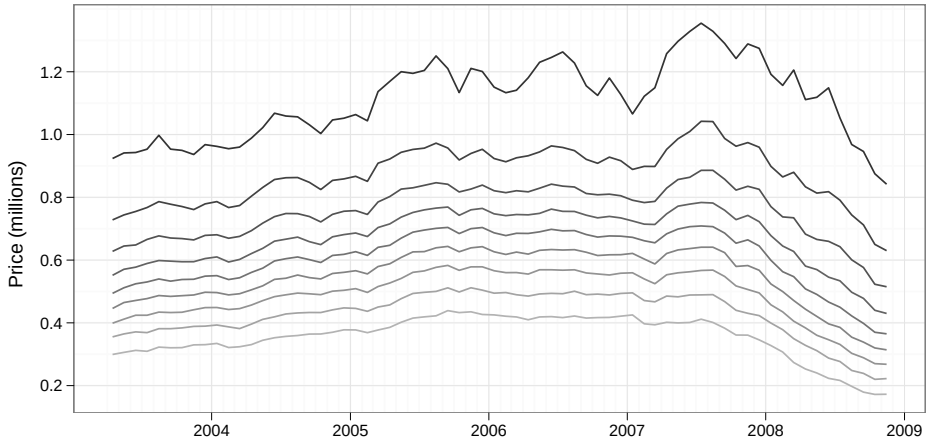

Figure 3: Average house price within each decile. Cheaper houses (lower deciles) are coloured lighter.

Another way to look at this inequality is Figure **??**. Here we have standardised the price of an "average" home to one. Relative to an average home, expensive houses have been getting more expensive and cheap houses cheaper.

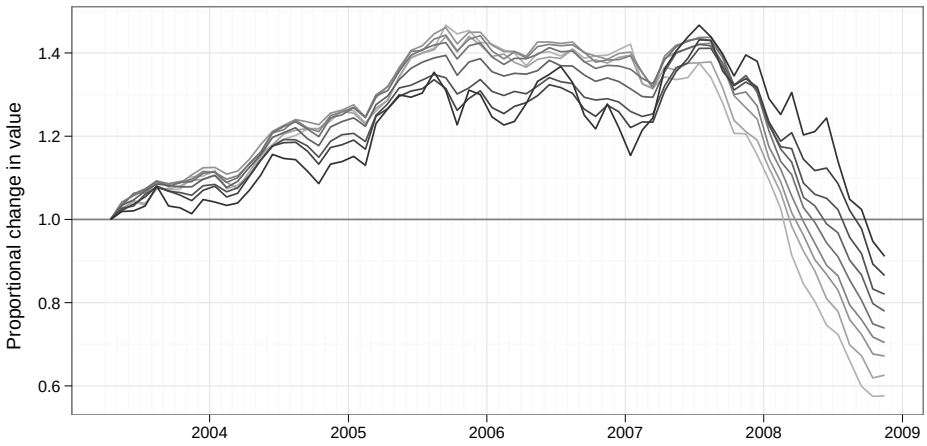

Figure 4: Indexed house price within each decile. The average price of cheaper houses increased more rapidly, and fell to much lower lows.

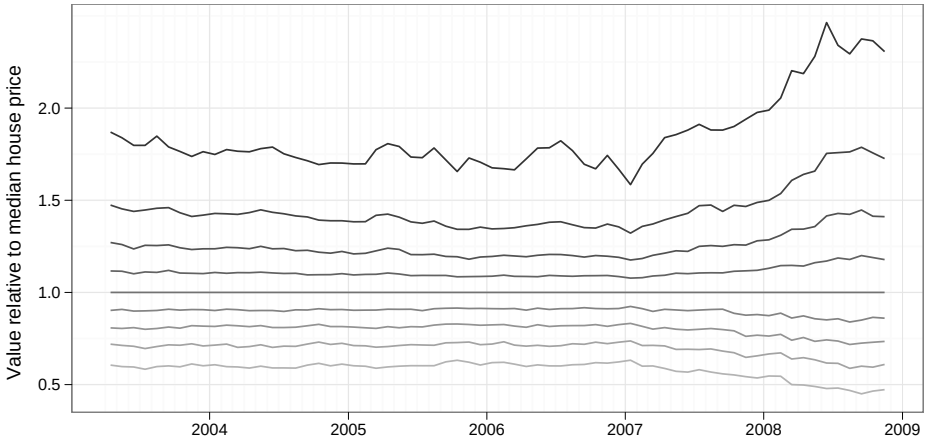

Figure 5: House prices, relative to the price of the median priced home. The disparity in home prices is increasing.

Geographic differences

Big city level break down: chose all cities with more than 2910 sales in total, an average of 10 sales per week. Gives us 59 cities in California with a total of 423,527 (out of an original 521,726, 81% of the original data).

From this reduced data we then calculated the number of sales and the average sale price for each city for each week. Figure 6 shows this data, with each city as a different line. Statisticians have an evocative name for this type of plot: the spaghetti plot. It's very hard to see anything in the big jumble of lines. We're going to use two techniques to bring some order to the mess. First, we're going to smooth the lines to focus on the overall trend, removing short-term variation that we're not so interested in. Next, we'll attempt to cluster the cities in groups that show similar patterns.

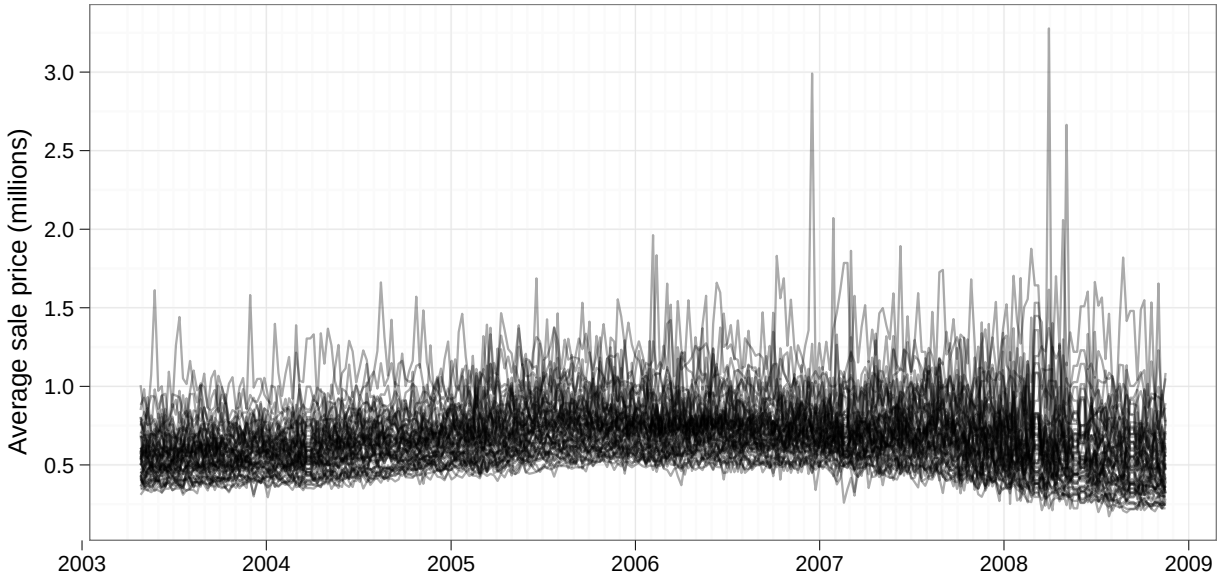

Figure 6: Average sale price for each week for each city. This type of plot is often called a spaghetti plot for obvious reasons

Generalised additive models (GAM) are a generalisation of linear models (Wood, 2006). A linear model has the basic form $y = a + b * x$, and GAMs generalise this to $y = f(x)$ where x is a smooth line. A linear model is special case of a GAM because lines are very smooth functions! Thin plate regression splines. We define wiggliness by the second derivative and then penalise it: the final result is a compromise between fit and wiggliness.

But don't worry if you don't understand the details: the gist is that GAMs are a useful way of removing noisy short-term effects and focussing on the general smooth trend. This is what we want: for this investigation, we're not interested in changes from week to week or day to day, we're interested in exploring the long term changes related to the housing crisis.

Displaying the smooth curves is an improvement, as in Figure 7. Note the big difference in scales: smoothing the data has removed the very large spikes which represent the sales of very expensive houses. But there are still a lot of the lines in that plot and we might be missing important findings. We'll make one other change to the data, we'll index each city: this removes the overall average price of the city and put's it onto an interpretable scale: proportional change in price since the start of the data.

Instead of displaying them all together, we display each city on a separate plot, as in Figure 8. This takes

up a lot of room, but if you have a big screen or a good printer it's very worthwhile. We can pick out some interesting patterns: San Francisco, Berkeley and Mountain view all show less of a peak and less of a drop: are more valuable than in 2004. Other cities show big peaks and big drops: Oakley, Vallejo, and San Pablo.

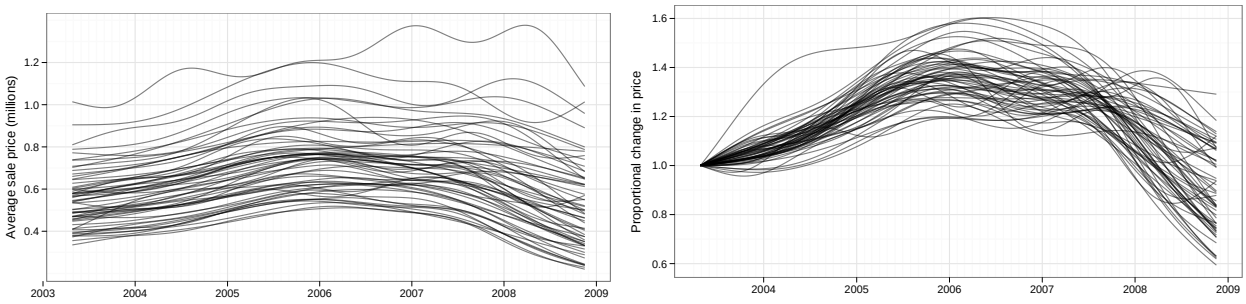

Figure 7: Smoothed city-level weekly average sale prices. Compared to the non-smoothed version it's easier to see the long-term trends, but it's still not particularly easy

After a few false starts we released that there were two main features that seemed to distinguish the different cities: the height of the peak in the boom and the most recent plummet. Figure 9 shows one way of clustering the cities into three groups. Note how the groups are basically formed along the diagonal: it's the difference between the peak and the plummet that seems to be telling us the most. Figure 10 shows the result of the clustering, with each of the three groups displayed in its own panel. The groups are somewhat arbitrary (we could shift the boundaries a little in either direction and have little effect)

Watching city growth

We can also use this data for an unexpected purpose. Because the data is so dense and includes the year that the house was built, we can explore historical patterns of housing development. This idea was inspired by truliaHindsight, `http://hindsight.trulia.com/`, but because we have the data in an unencumbered form we are much freer to experiment with the visualisation of this data. They have much more data, 150,000 vs 27,000, but we can display more (they display at most 2000 points at once), and we can do more sophisticated analyses.

We'll overlay the data on a map of San Francisco from openstreetmap.

Histogram of year built.

You can see features like golden gate park and so on.

Conclusions

All the tools we use are open source: you can download them and replicate our work yourselves. The principle of reproducible research (cite Gentleman and Temple Lang) paper is very important for science - we provide enough detail that you can follow out work every step of the way, and you can run a script to reproduce exactly what we did. A little tricky because working on this data analysis also lead us to develop some new tools, and it takes some time for these to trickle into released versions. If you have problems running the code we released, please let us know! Data analysis like software development. Local caches to speed things up and to provide some backup if the original sources go down.

Tension with interactive tools: they are great for discovery, but bad for reproducibility. Once you have discovered something in your interactive tool, you need to be able to reproduce it independently so that others can see it too. Area of active research (cite Heer's work). Also need to note your findings as you go

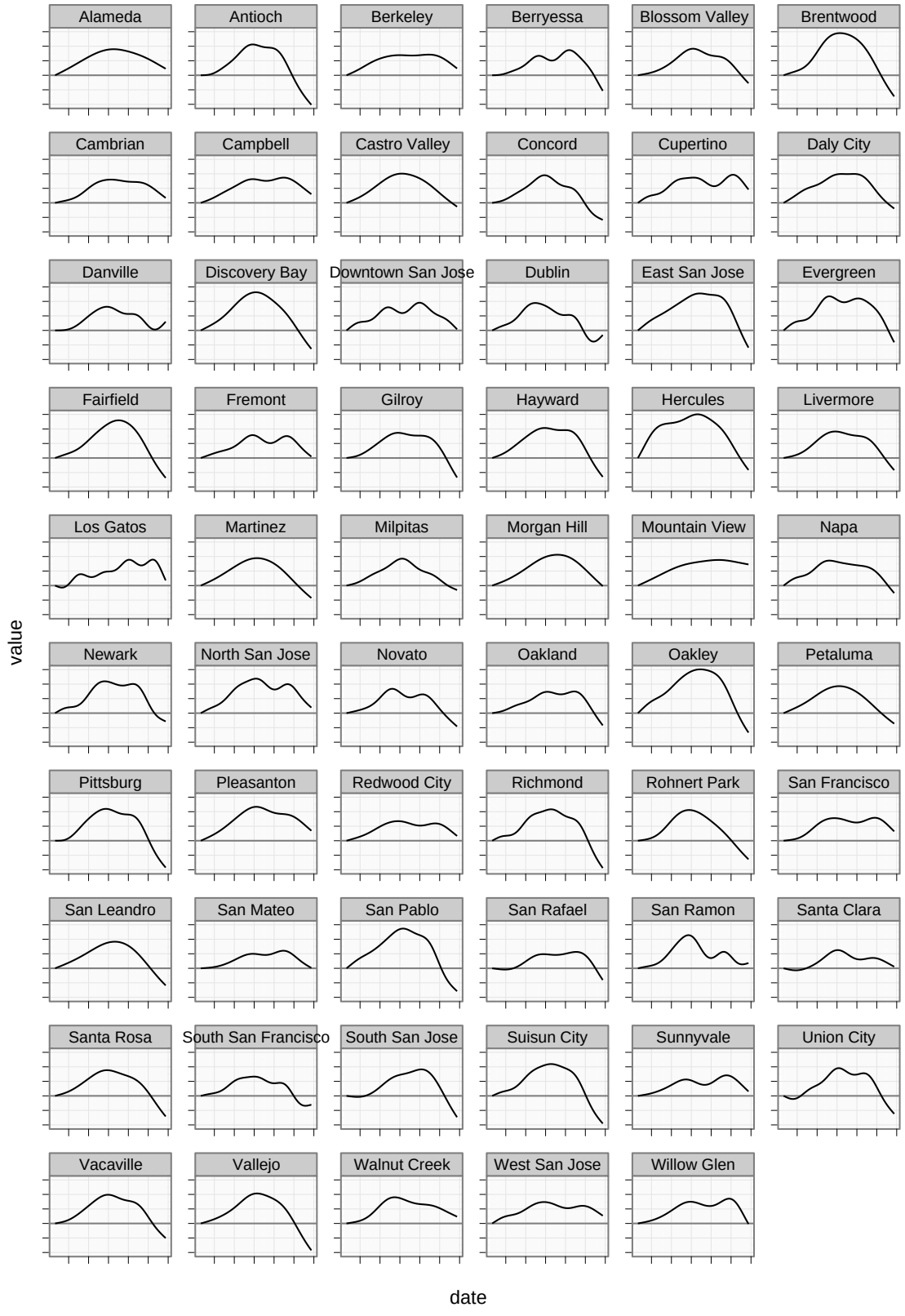

Figure 8: Individual plots for each city. Axis labels have been removed to save space, but the same limits are used for each plot.

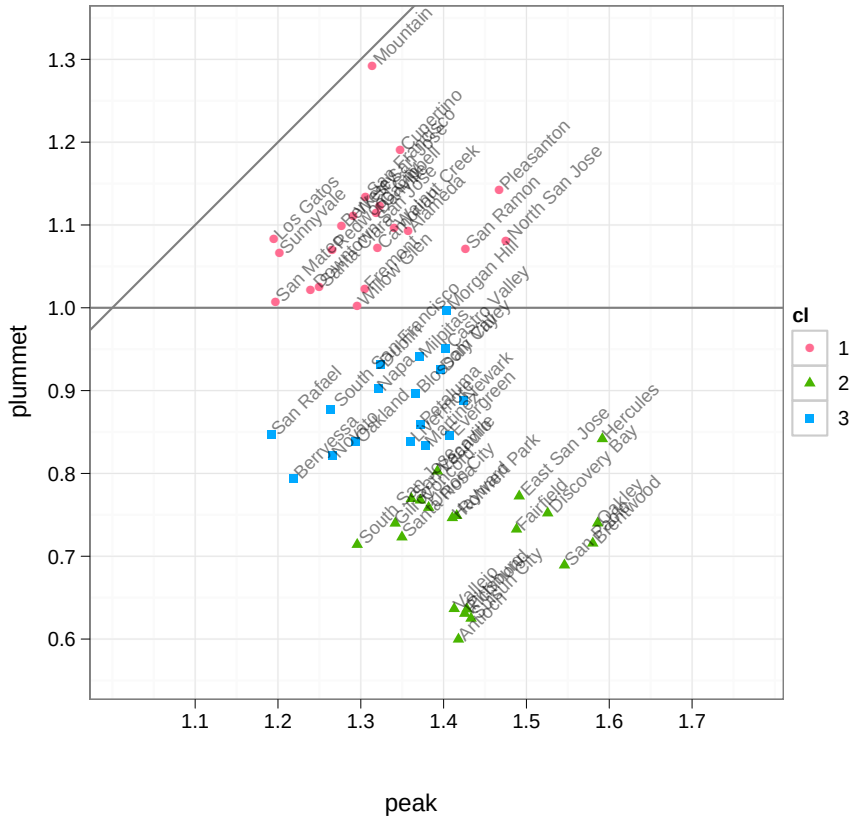

Figure 9: A scatterplot of cities with height at peak on x axis, and most recent value (plummet) on y axis. The cities have been clustered into three groups. The closer a city is to the line in the top left corner, the less the effect of the housing crisis on average prices.

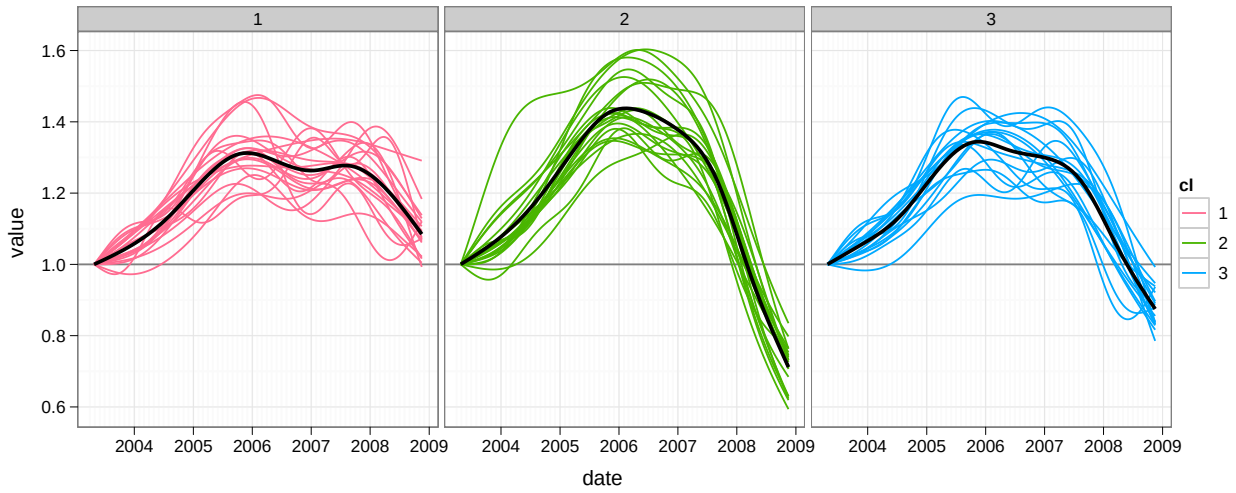

Figure 10: The index sale price time series for the three clustered identified in Figure 9. Cluster one includes cities with low peak and no plummet, cluster two cities with high peak and big plummet, and cluster three is somewhere in between. The thick lines represent smoothed patterns within each cluster.

along - no way to do this purely in code. If you read the code on the website you'll see we've used comments to note down what we see and the analysis follows a fairly logical flow. This is different to what happens in practice - there are many blind alleys that didn't make the final cut. We rely on the rcs system to keep these, although currently lacking tools to easily search past versions and see the wrong paths that we went down.

Tools: shell (wget, awk); R (ggplot2, plyr, reshape).

Note about graphics: can churn out rough versions for exploration very quickly - takes more time to polish them for publication. Clarify story, remove extraneous elements and ensure that it supports the text.

References

Simon Wood. *Generalized Additive Models: An Introduction with R*. Chapman Hall/CRC, 2006.