

# polyr

## Modelling large data

Hadley Wickham

1. Strategy for analysing large data.
2. Introduction to the Texas housing data.
3. What's happening in Houston?
4. Using a models as a tool
5. Using models in their own right

# Large data strategy

Start with a single unit, and identify interesting patterns.

Summarise patterns with a model.

Apply model to all units.

Look for units that don't fit the pattern.

Summarise with a single model.

# Texas housing data



For each metropolitan area (45) in Texas,  
for each month from 2000 to 2009 (112):

Number of houses listed and sold

Total value of houses, and average sale  
price

Average time on market

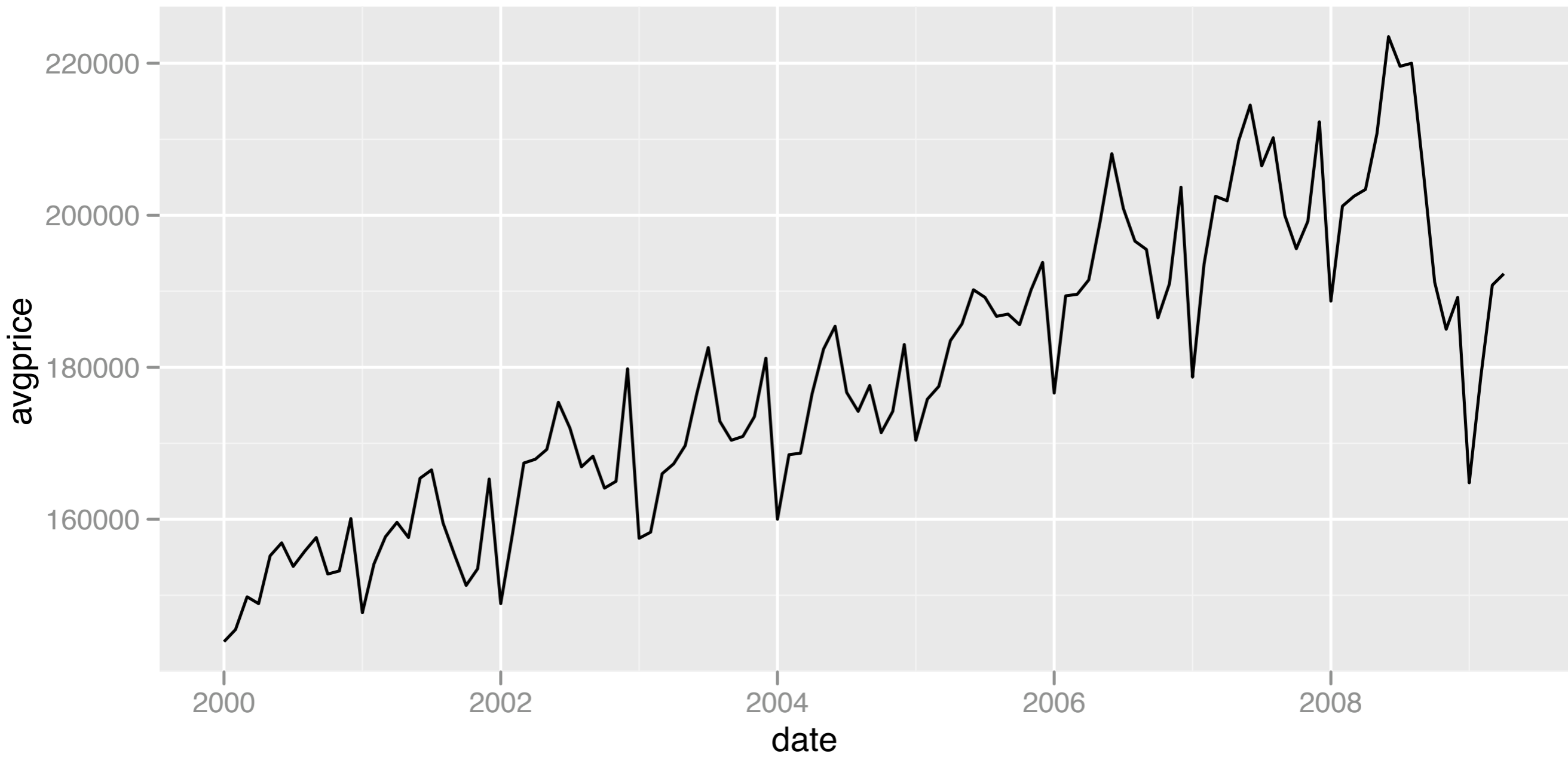


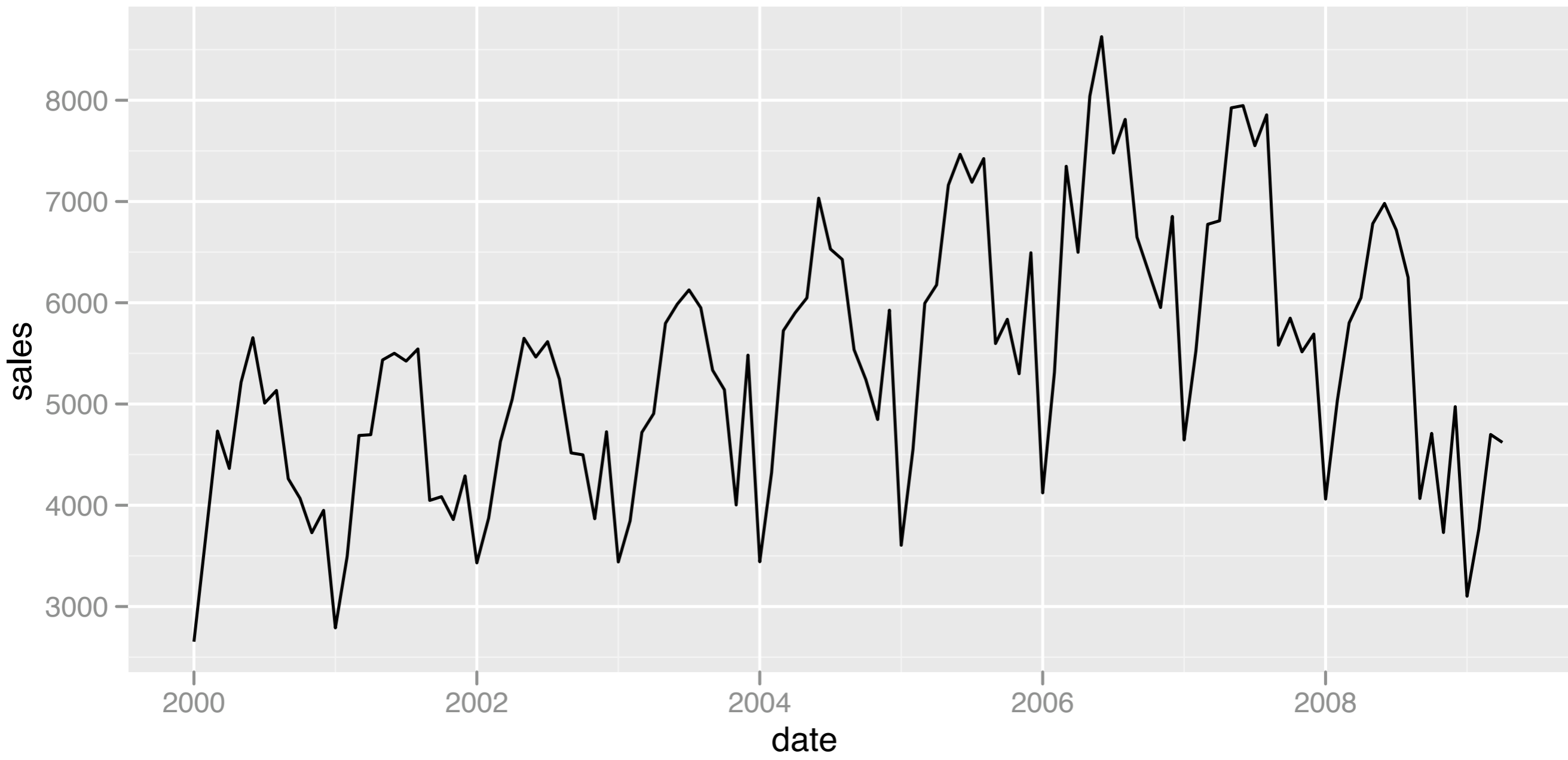
# Strategy

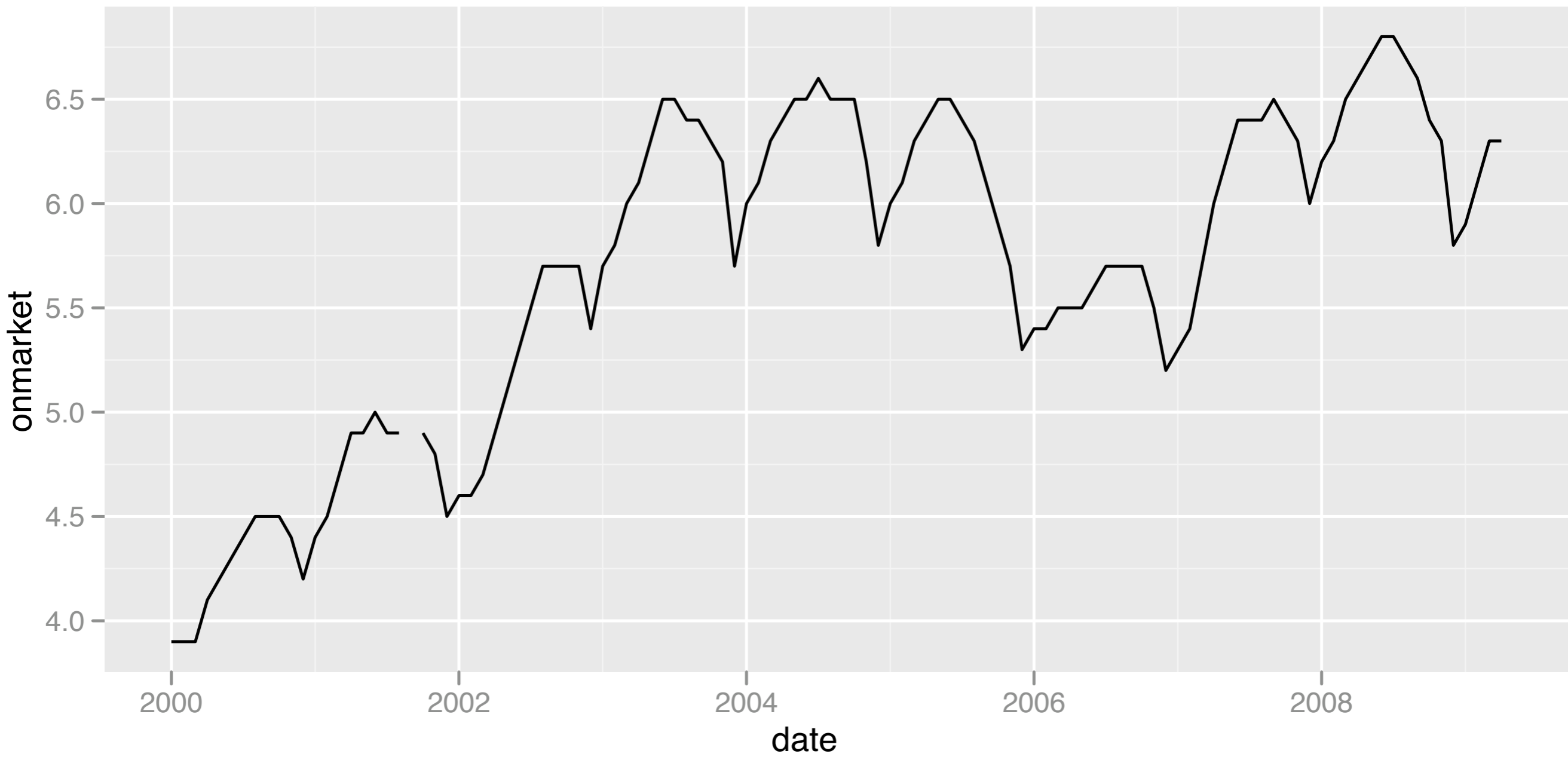
Start with a single city (Houston).

Explore patterns & fit models.

Apply models to all cities.





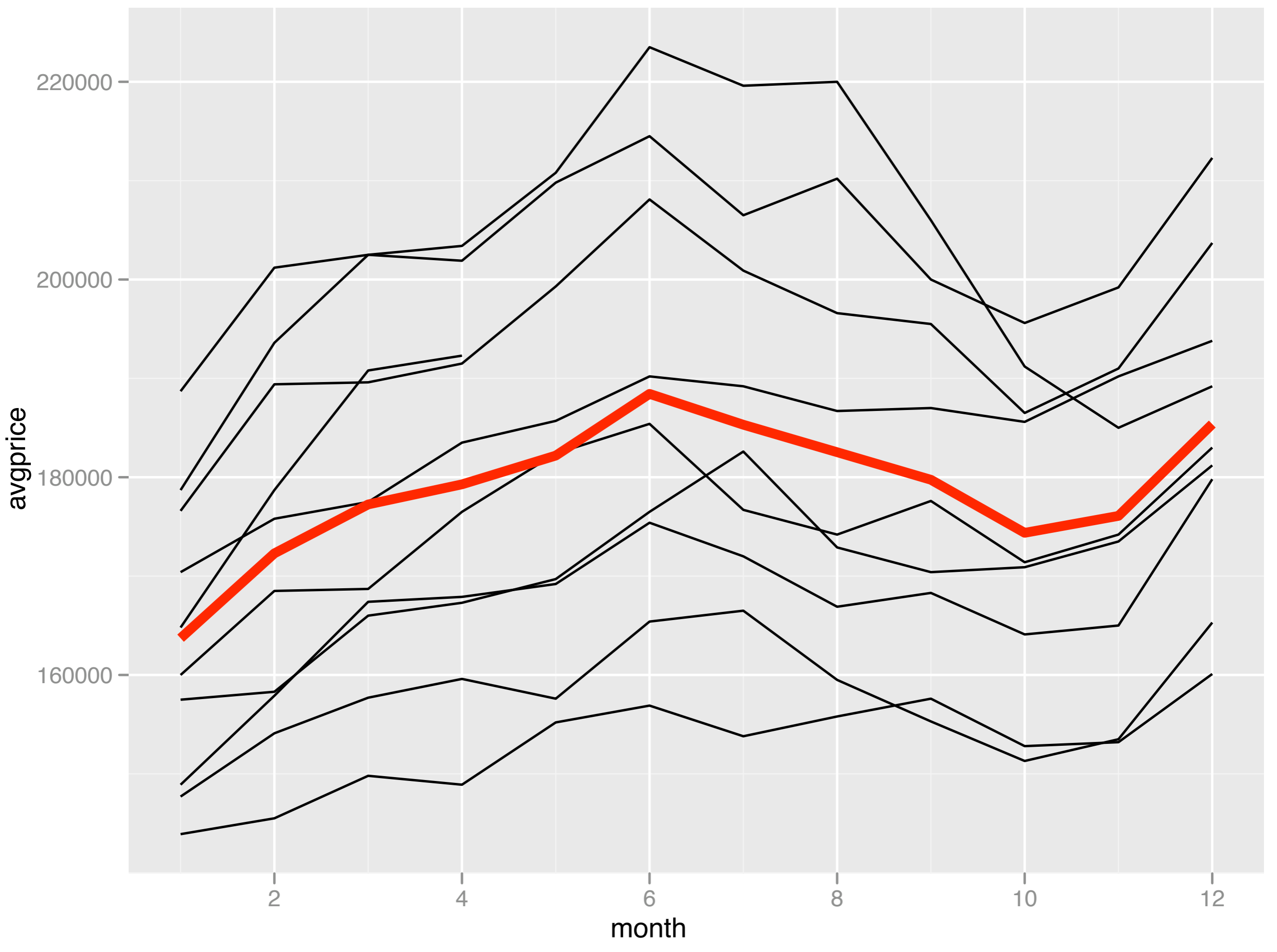




# Seasonal trends

Make it much harder to see long term trend. How can we remove the trend?

(Many sophisticated techniques from time series, but what's the simplest thing that might work?)



# Challenge

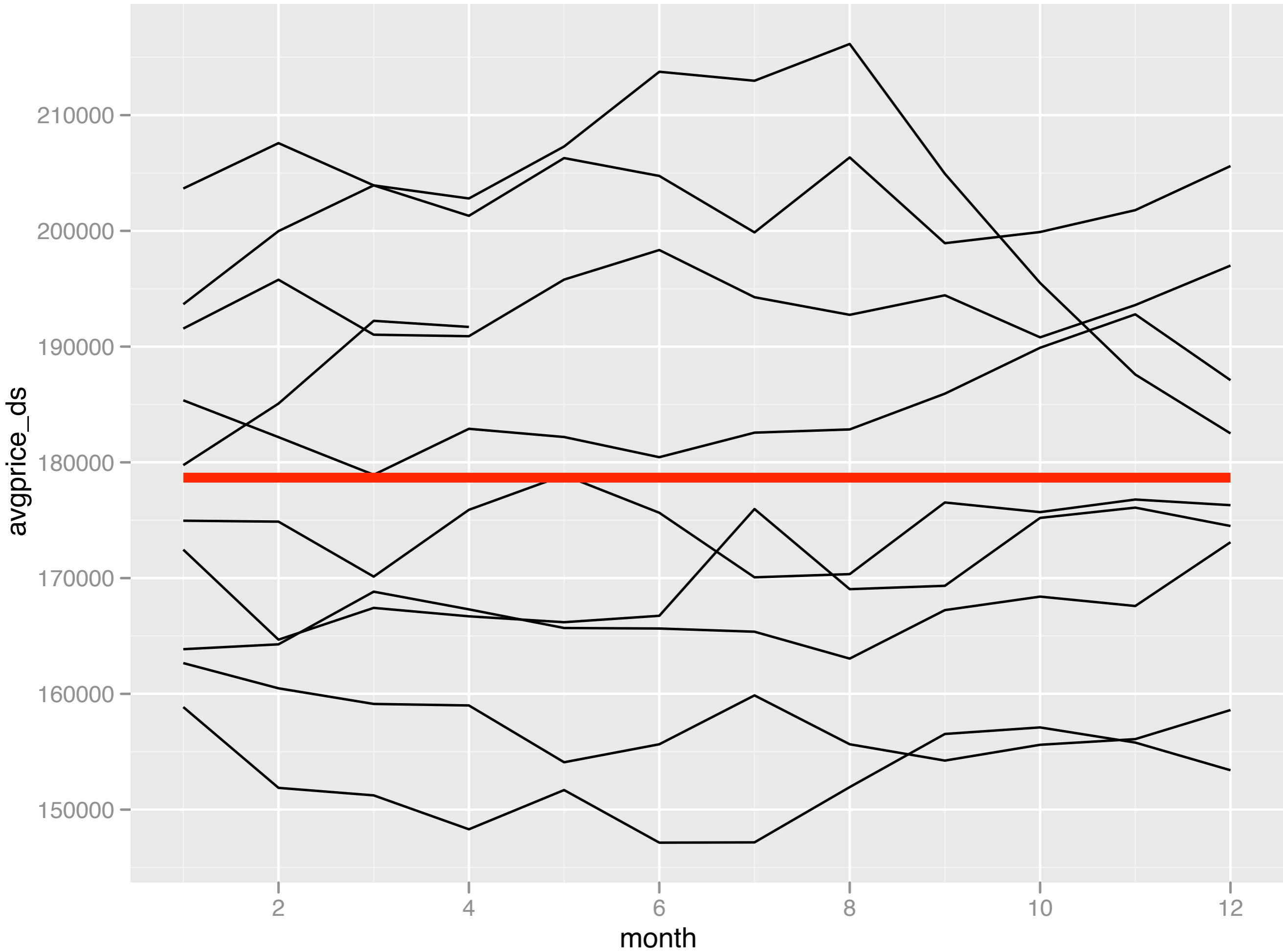
What does the following function do?

```
deseas <- function(var, month) {  
  resid(lm(var ~ factor(month))) +  
  mean(var, na.rm = TRUE)  
}
```

How could you use it in conjunction with transform to deasonalise the data? What if you wanted to deasonalise every city?

```
houston <- transform(houston,  
  avgprice_ds = deseas(avgprice, month),  
  listings_ds = deseas(listings, month),  
  sales_ds = deseas(sales, month),  
  onmarket_ds = deseas(onmarket, month)  
)
```

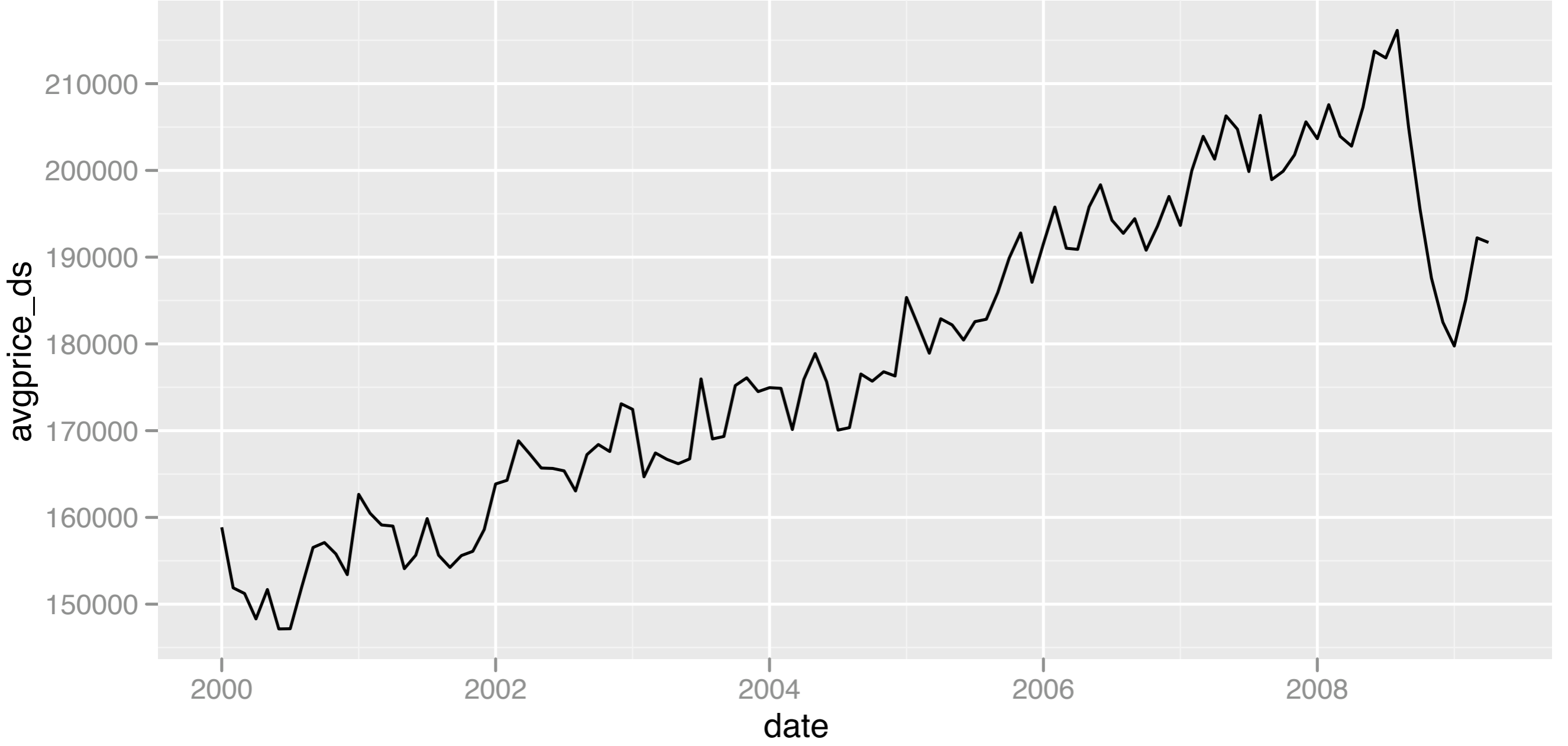
```
qplot(month, sales_ds, data = houston,  
  geom = "line", group = year) + avg
```

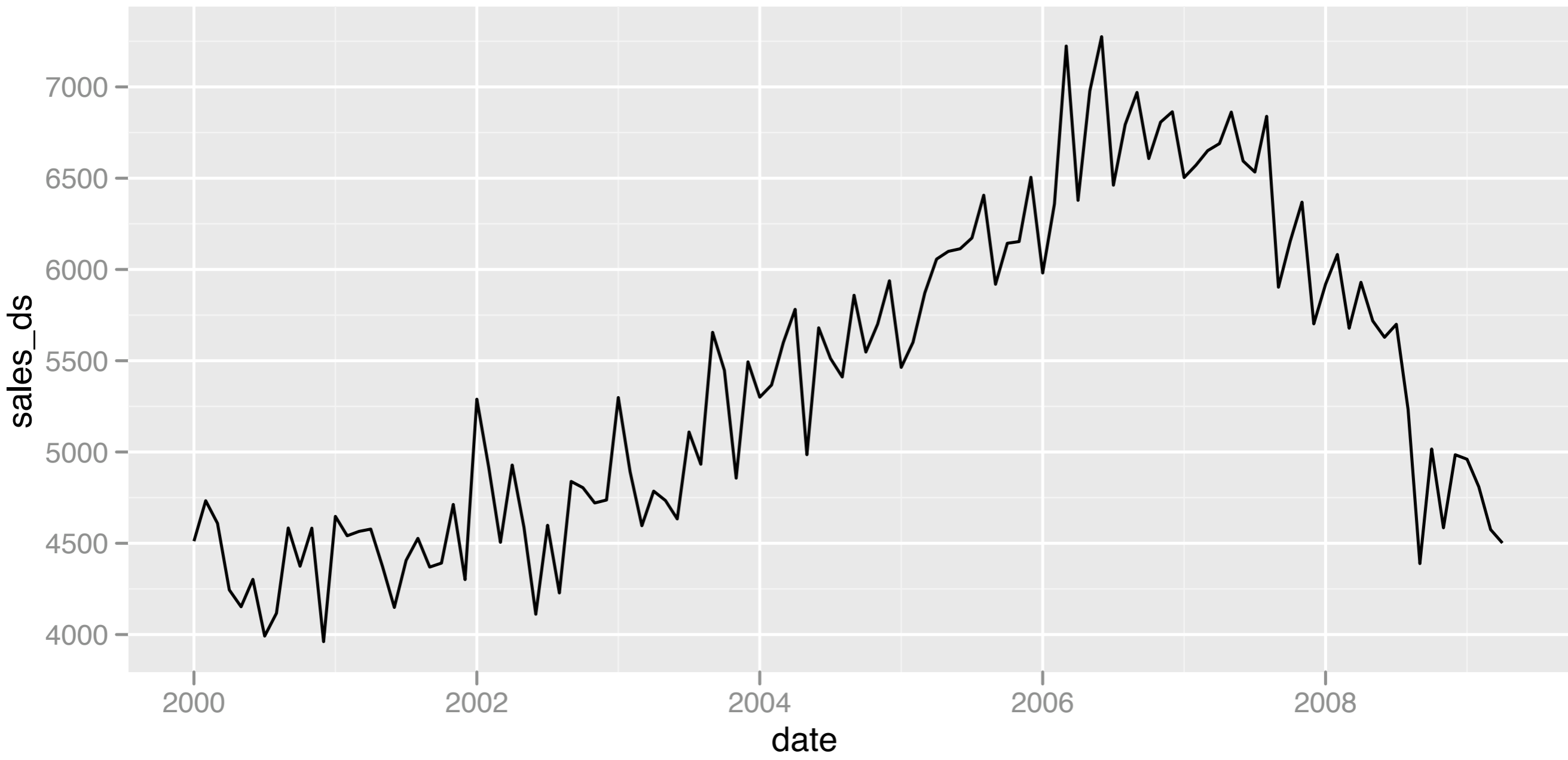


# Model as tools

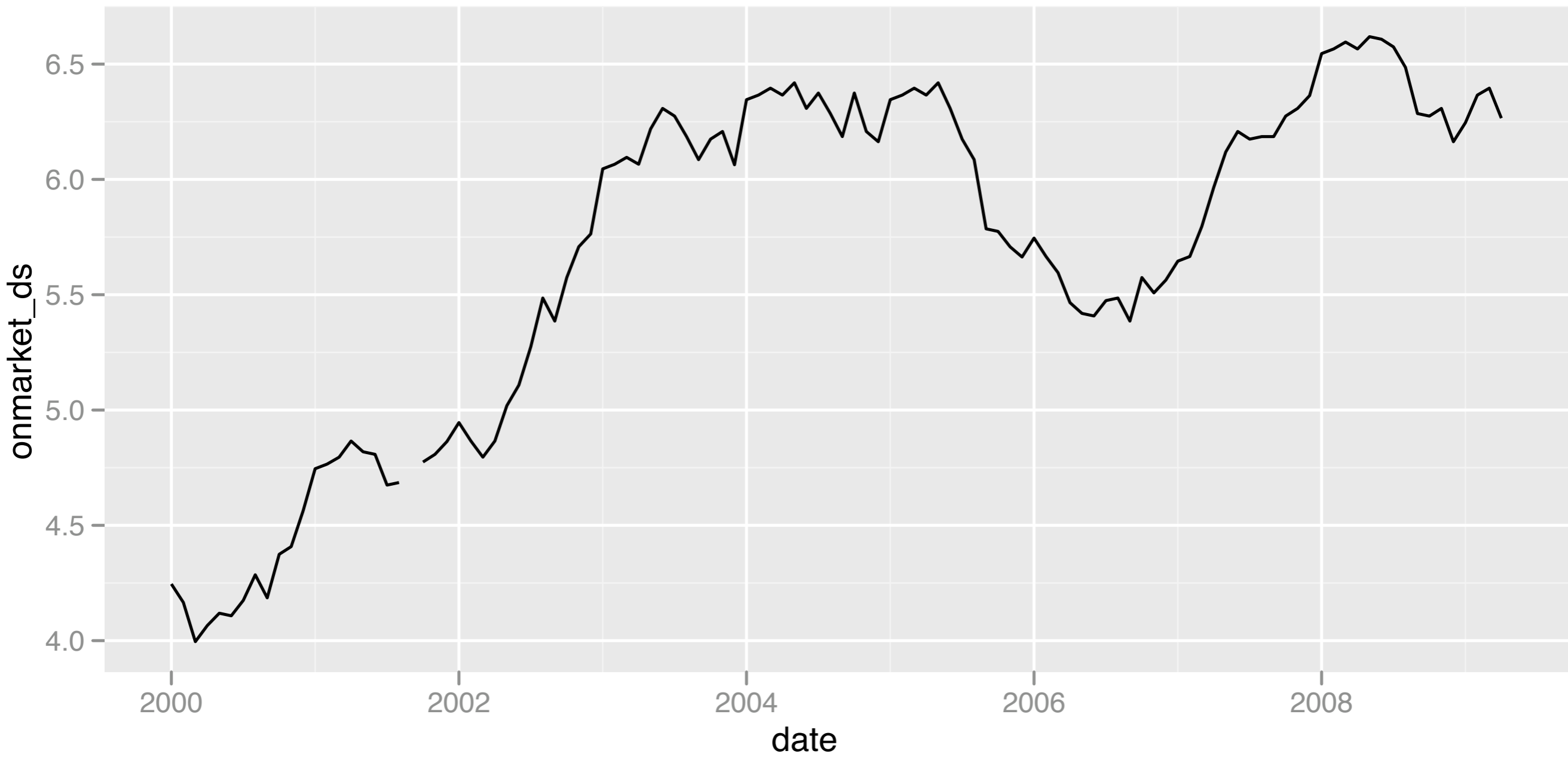
Here we're using the linear model as a tool - we don't care about the coefficients or the standard errors, just using it to get rid of a striking pattern.

Tukey described this pattern as residuals and reiteration: by removing a striking pattern we can see more subtle patterns.





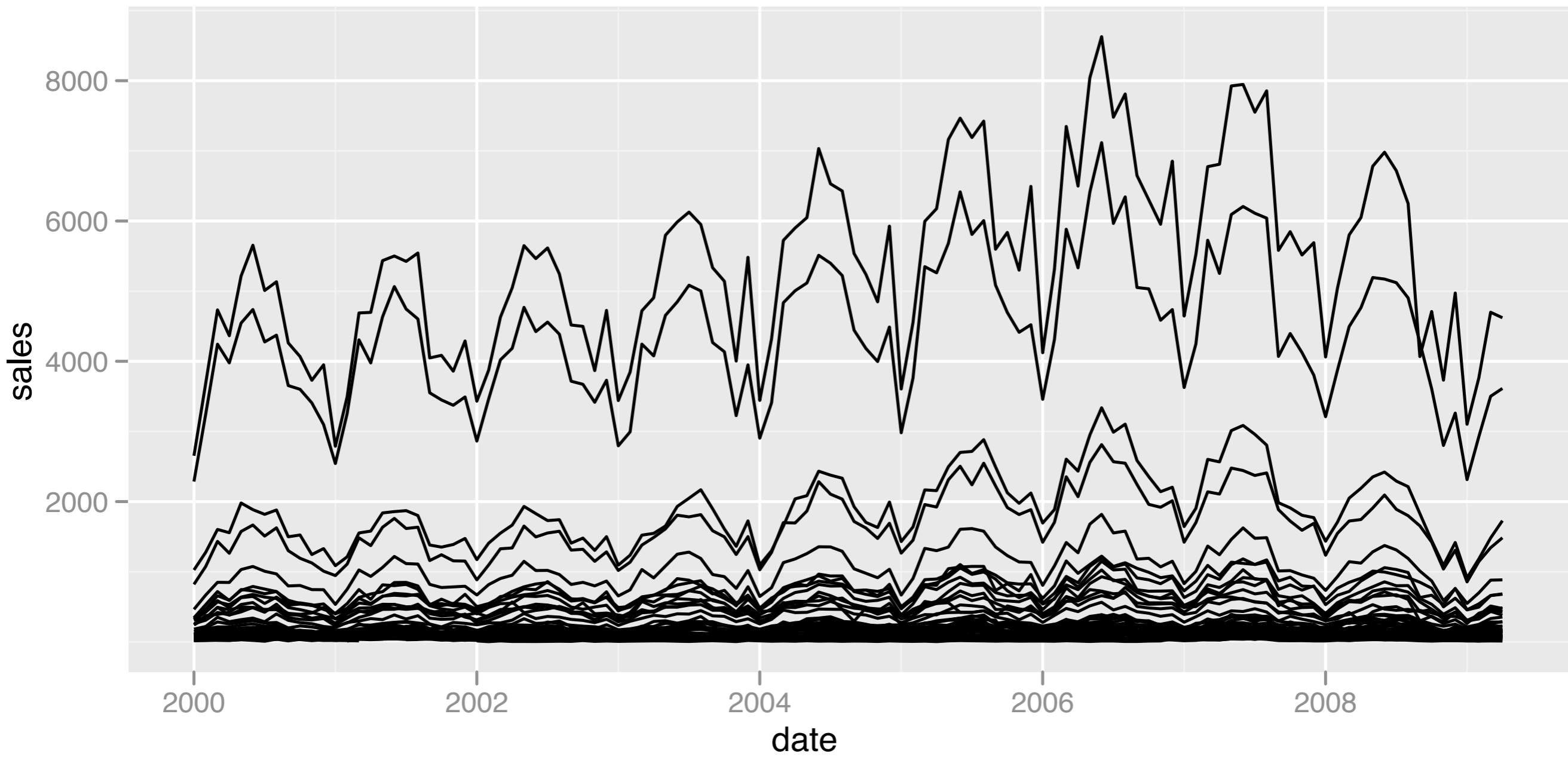




# Summary

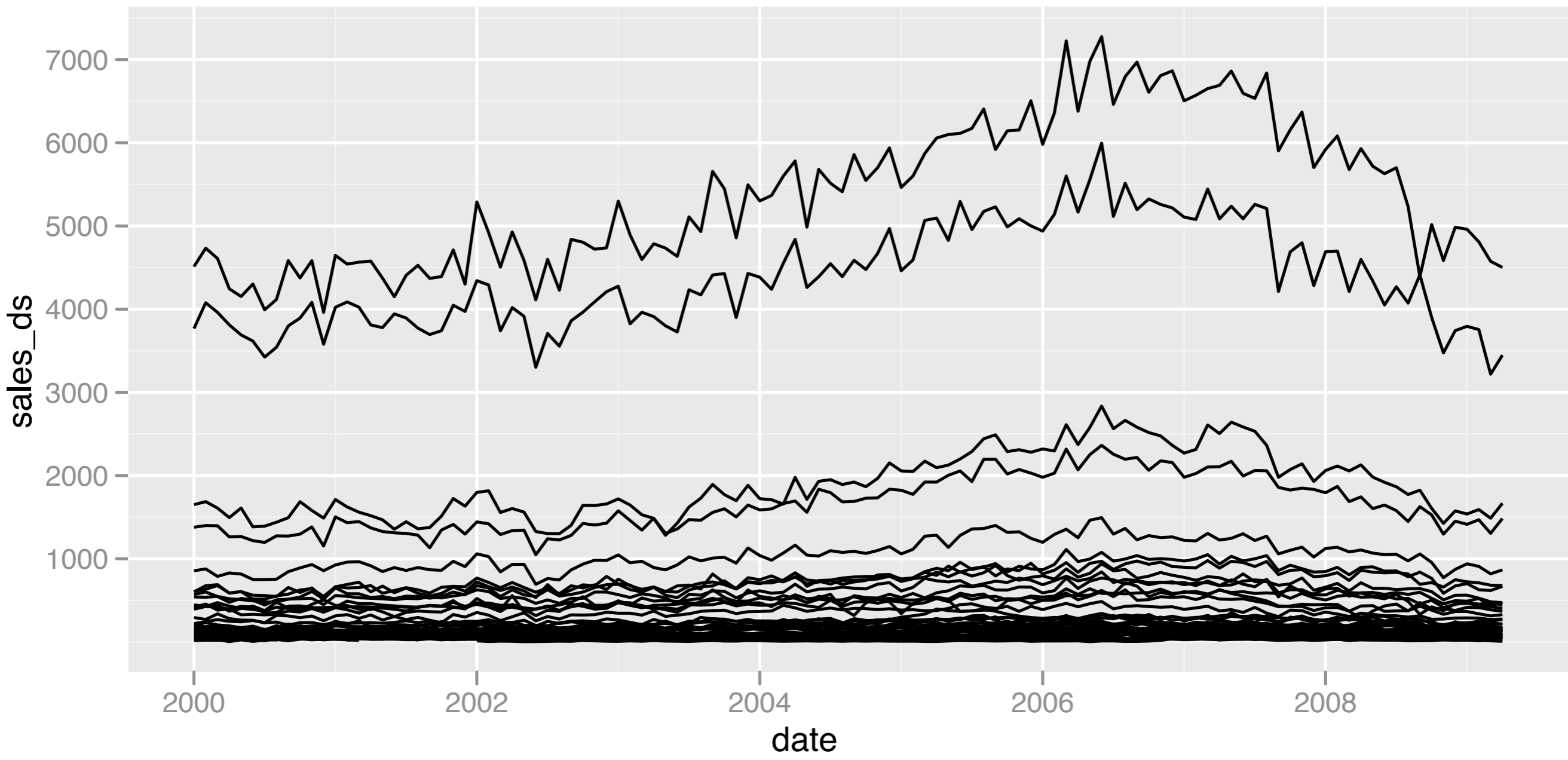
Most variables seem to be combination of strong seasonal pattern plus weaker long-term trend.

How do these patterns hold up for the rest of Texas? We'll focus on sales.



```
tx <- read.csv("tx-house-sales.csv")
qplot(date, sales, data = tx, geom = "line",
       group = city)

tx <- ddpoly(tx, "city", transform,
            sales_ds = deseas(sales, month))
qplot(date, sales_ds, data = tx, geom = "line",
       group = city)
```



# It works, but...

It doesn't give us any insight into the similarity of the patterns across multiple cities. Are the trends the same or different?

So instead of throwing the models away and just using the residuals, let's keep the models and explore them in more depth.

# Two new tools

`dply`: takes a data frame, **splits** up in the same way as `ddply`, **applies** function to each piece and **combines** the results into a list

`ldply`: takes a list, **splits** up into elements, **applies** function to each piece and then **combines** the results into a data frame

`dply` + `ldply` = `ddply`

```
models <- dply(tx, "city", function(df)
  lm(sales ~ factor(month), data = df))
```

```
models[[1]]
coef(models[[1]])
```

```
ldply(models, coef)
```



# Labelling

Notice we didn't have to do anything to have the coefficients labelled correctly.

Behind the scenes plyr records the labels used for the split step, and ensures they are preserved across multiple plyr calls.

# Back to the model

What are some problems with this model?  
How could you fix them?

Is the format of the coefficients optimal?

Turn to the person next to you and  
discuss for 2 minutes.

```
qplot(date, log10(sales), data = tx, geom = "line",  
group = city)
```

```
models2 <- dply(tx, "city", function(df)  
  lm(log10(sales) ~ factor(month), data = df))
```

```
coef2 <- lapply(models2, function(mod) {  
  data.frame(  
    month = 1:12,  
    effect = c(0, coef(mod)[-1]),  
    intercept = coef(mod)[1])  
})
```

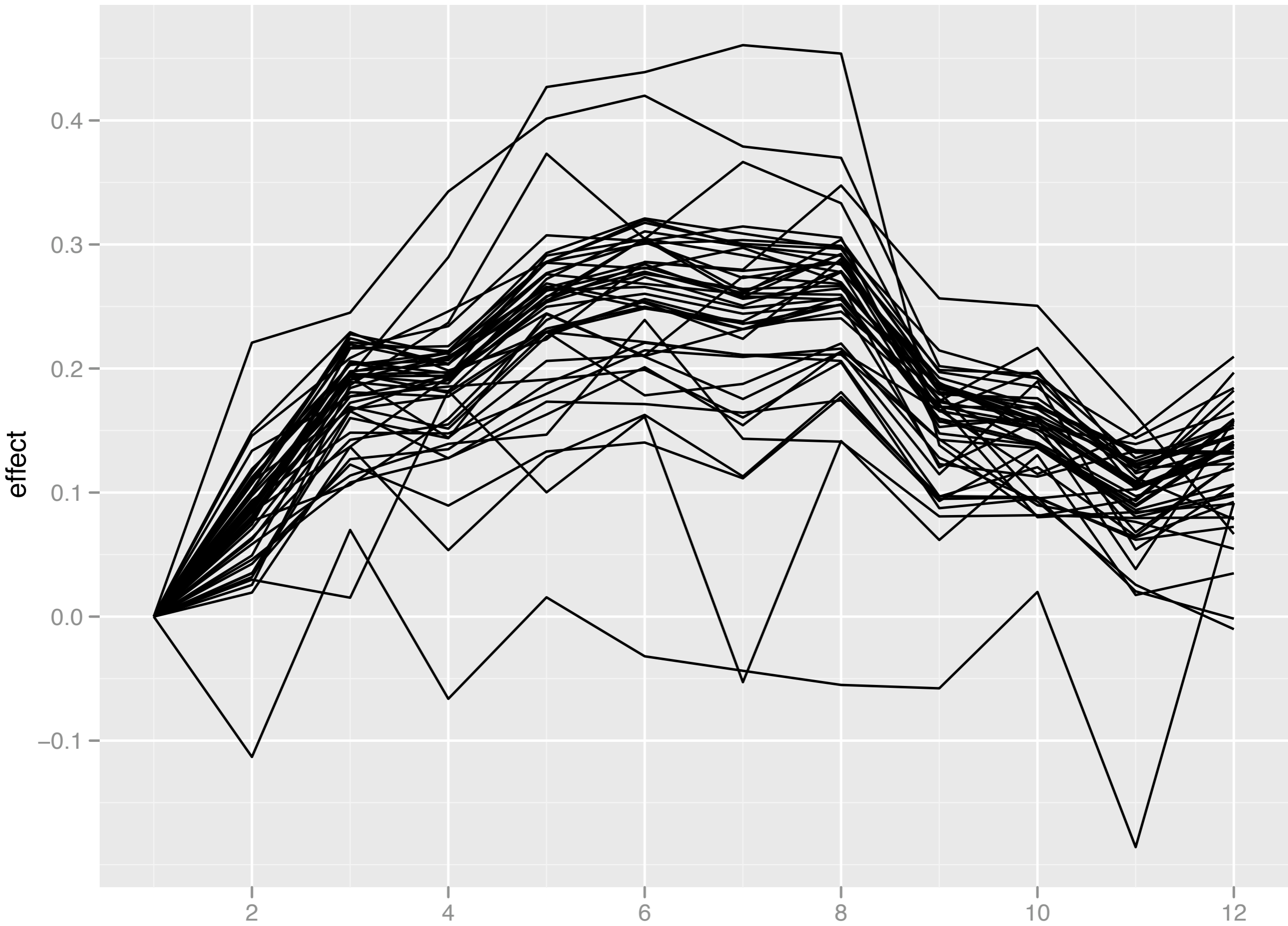
```
qplot(date, log10(sales), data = tx, geom = "line",  
group = city)
```

Log transform sales to  
make coefficients  
comparable (ratios)

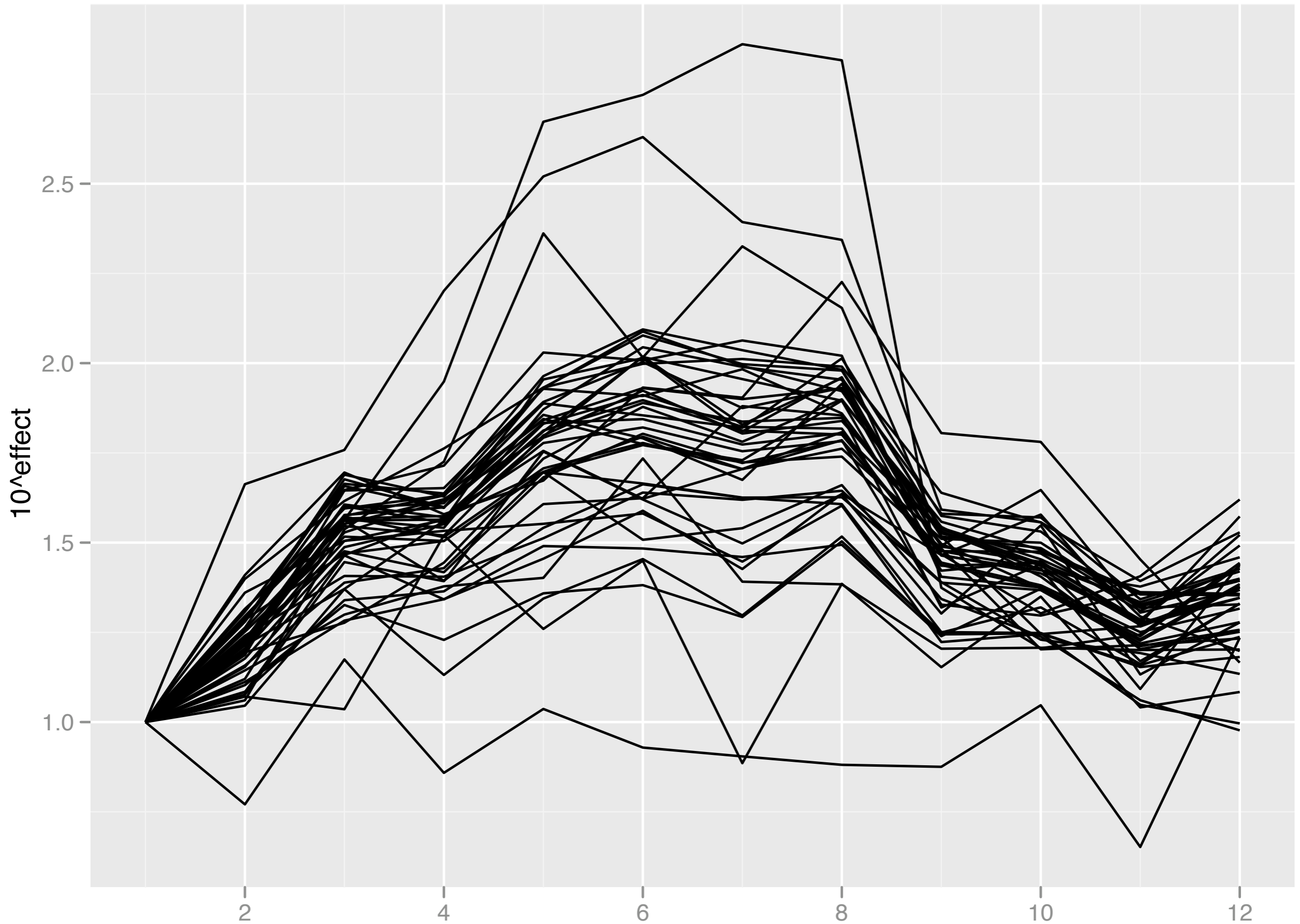
```
models2 <- dplyr::group_by(tx, "city") %>%  
  summarise(mod = lm(log10(sales) ~ factor(month), data = df))
```

```
coef2 <- lapply(models2, function(mod) {  
  data.frame(  
    month = 1:12,  
    effect = c(0, coef(mod)[-1]),  
    intercept = coef(mod)[1])  
})
```

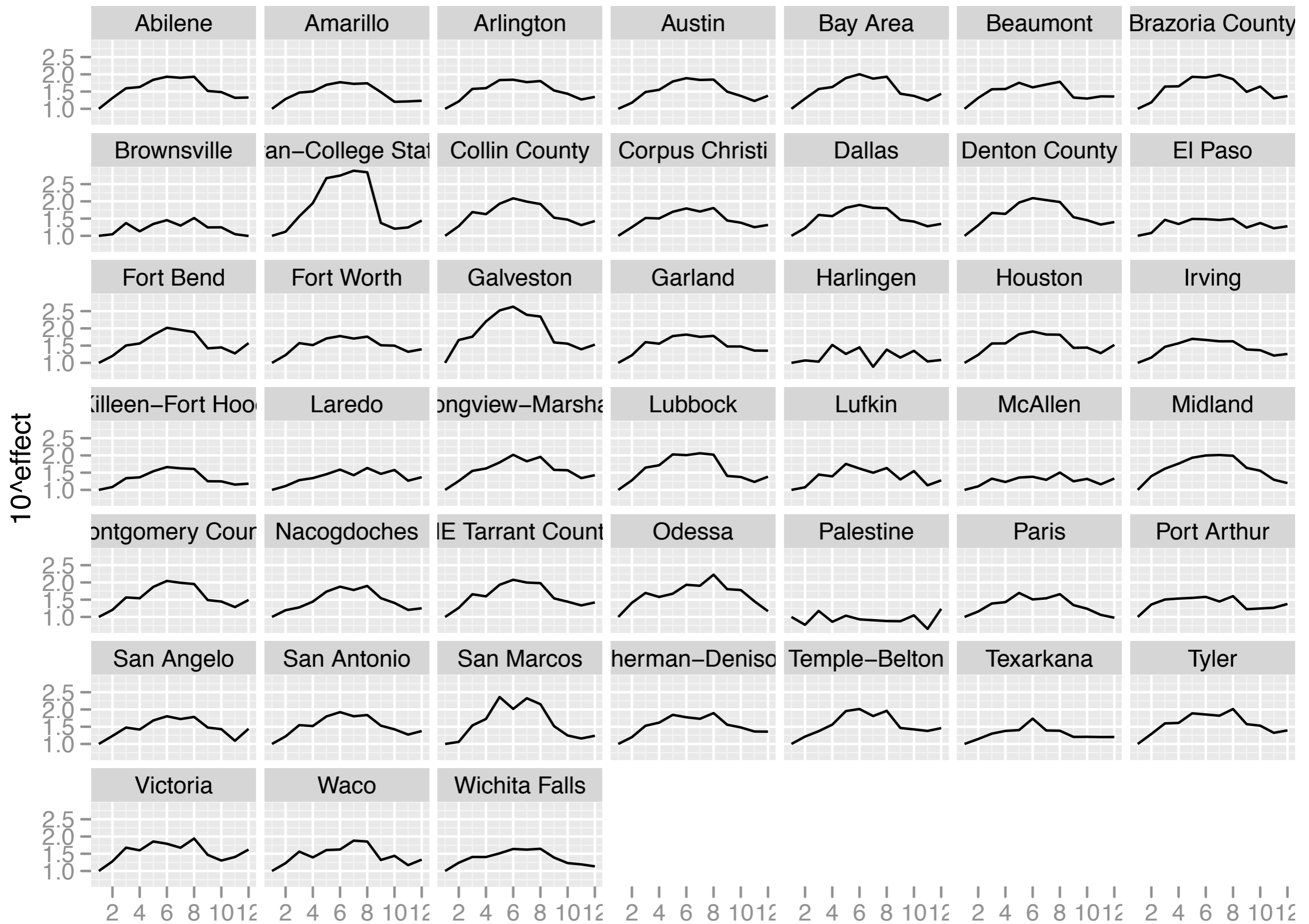
Puts coefficients in  
rows, so they can be  
plotted more easily



```
qplot(month, effect, data = coef2, group = city, geom = "line")
```



```
qplot(month, 10 ^ effect, data = coef2, group = city, geom = "line")
```



```
qplot(month, 10 ^ effect, data = coef2, geom = "line") + facet_wrap(~ city)
```

# What should we do next?

What do you think?

You have 30 seconds to come up with (at least) one idea.



# My ideas

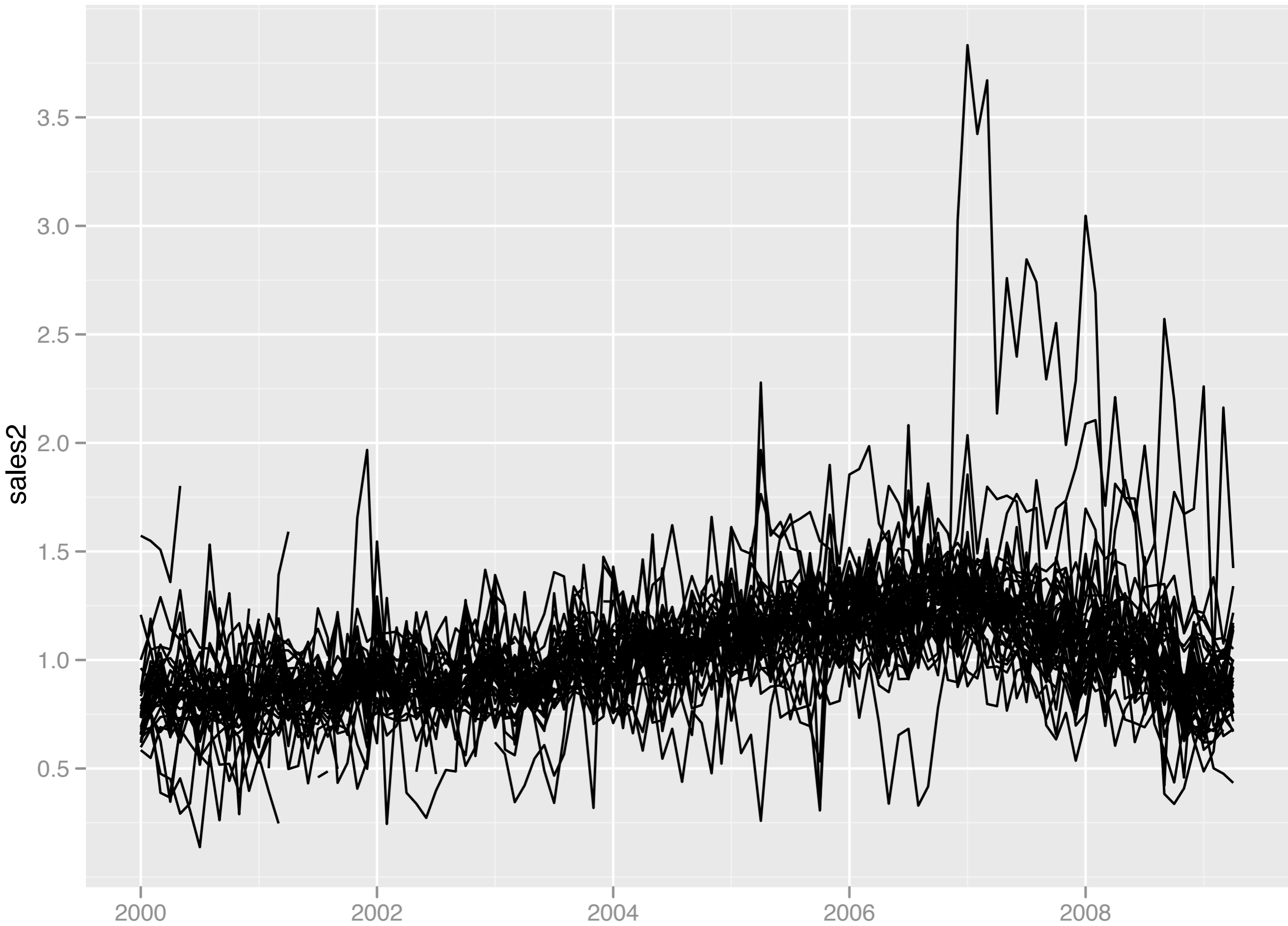
Fit a single model,  $\log(\text{sales}) \sim \text{city} * \text{factor}(\text{month})$ , and look at residuals

Fit individual models,  $\log(\text{sales}) \sim \text{factor}(\text{month}) + \text{ns}(\text{date}, 3)$ , look cities that don't fit

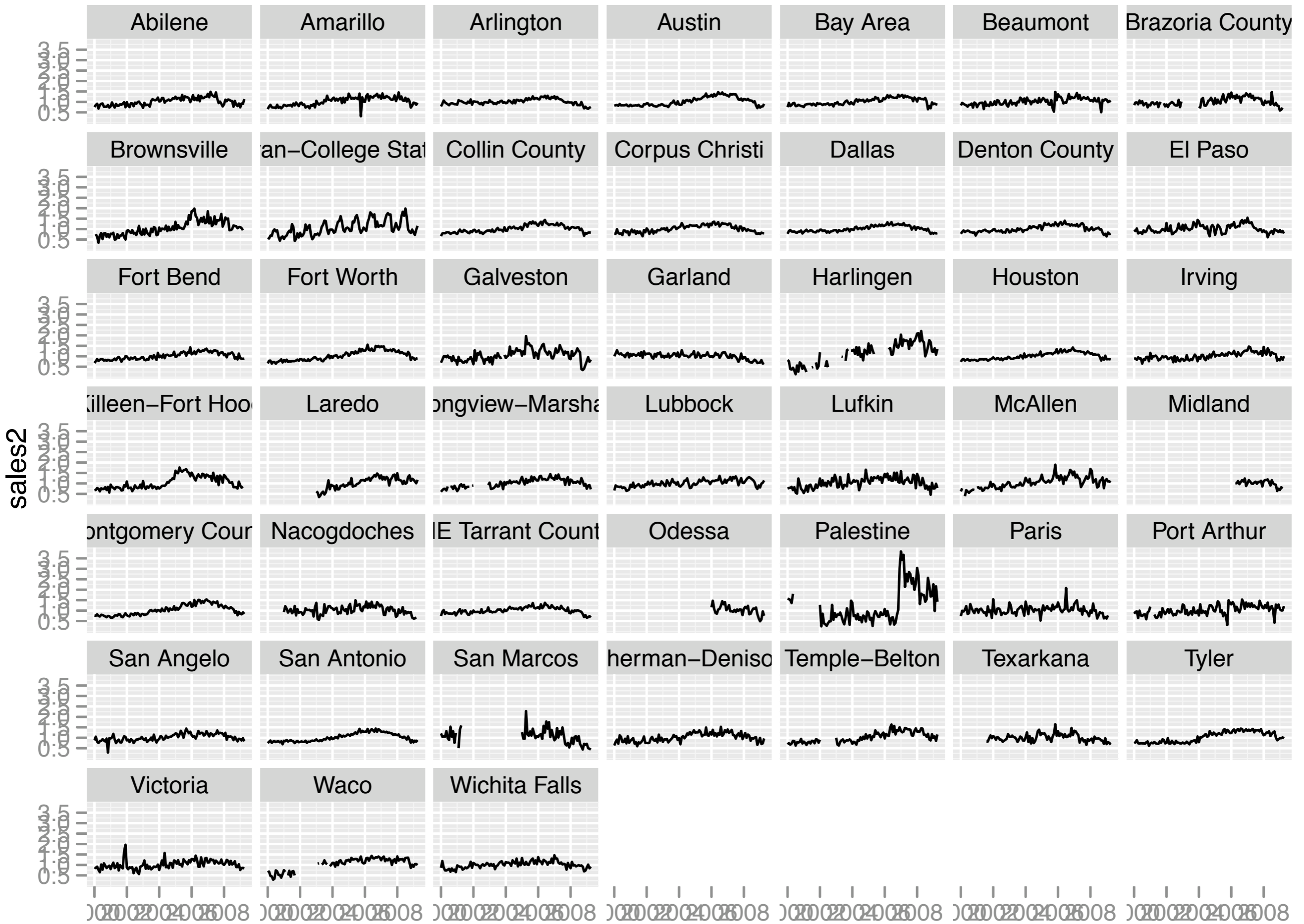
```
# One approach - fit a single model

mod <- lm(log10(sales) ~ city + factor(month),
  data = tx)

tx$sales2 <- 10 ^ resid(mod)
qplot(date, sales2, data = tx, geom = "line",
  group = city)
last_plot() + facet_wrap(~ city)
```



```
qplot(date, sales2, data = tx, geom = "line", group = city)
```



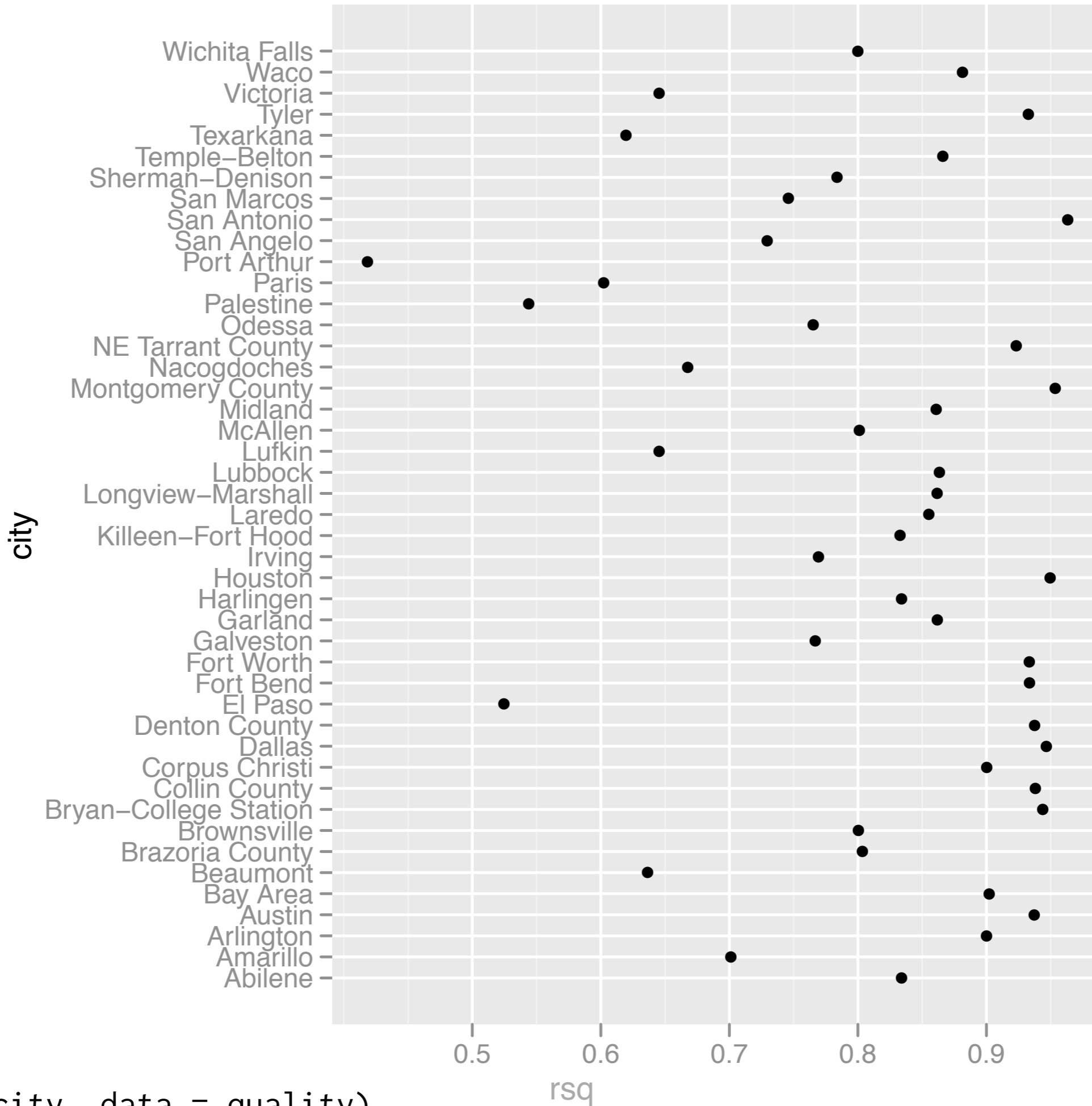
```
last_plot() + facet_wrap(~ city)
```

date

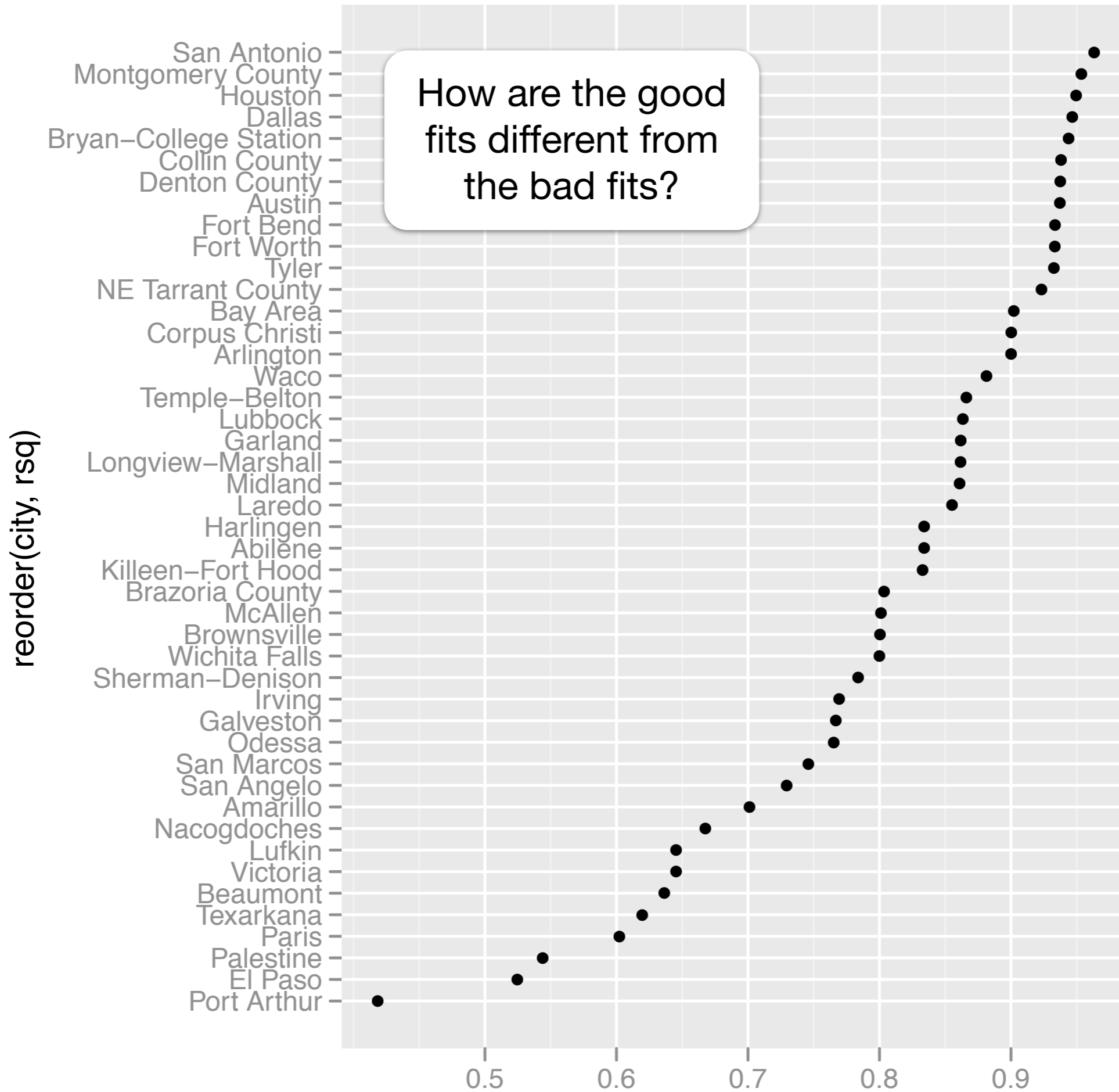
```
# Another approach: Essence of most cities is seasonal
# term plus long term smooth trend. We could fit this
# model to each city, and then look for models which don't
# fit well.

library(splines)
models3 <- dply(tx, "city", function(df) {
  lm(log10(sales) ~ factor(month) + ns(date, 3), data = df)
})

# Extract rsquared from each model
rsq <- function(mod) c(rsq = summary(mod)$r.squared)
quality <- ldply(models3, rsq)
```



qplot(rsq, city, data = quality)



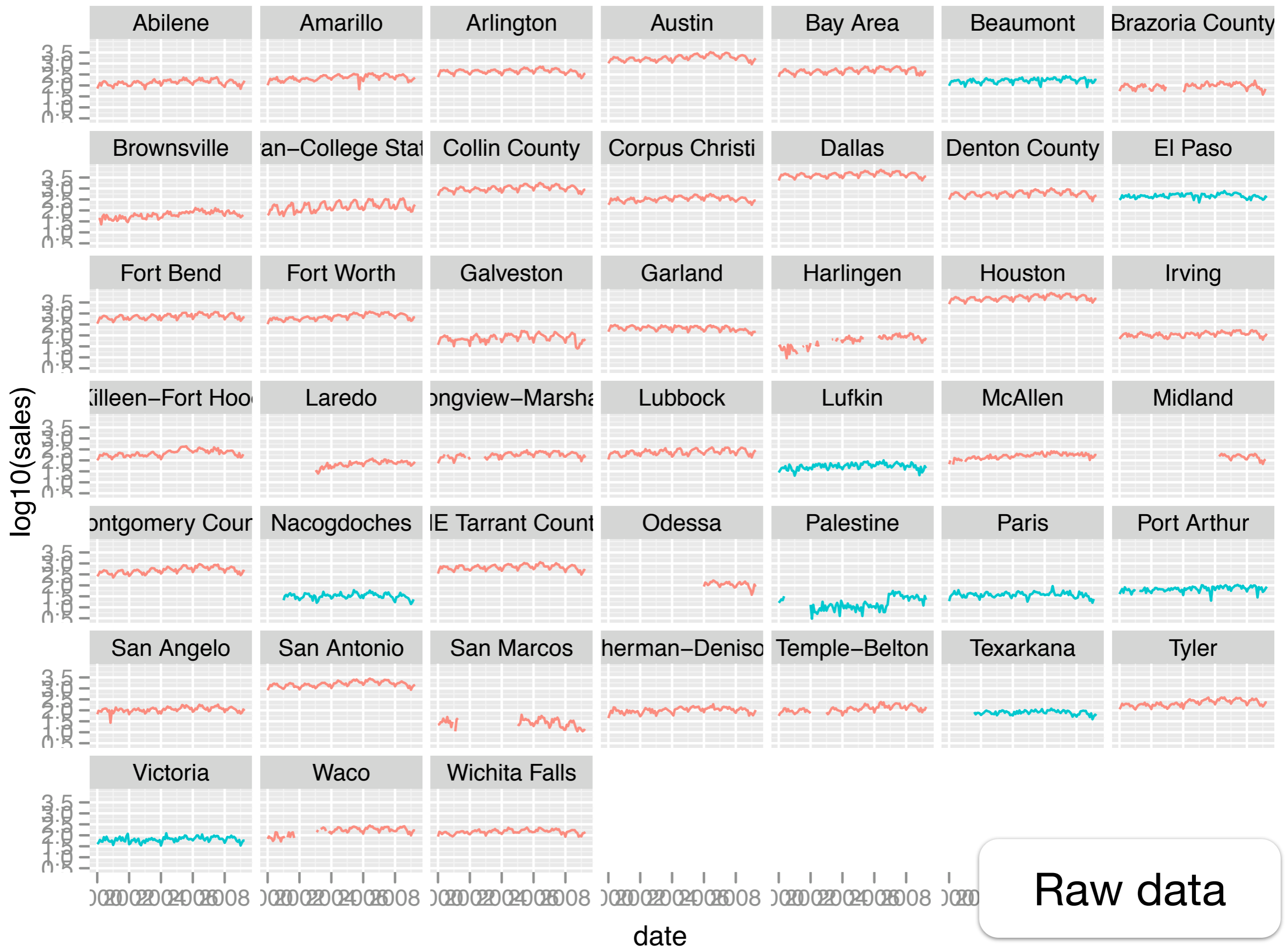
```
qplot(rsq, reorder(city, rsq), data = quality)
```

```
quality$poor <- quality$rsq < 0.7
tx2 <- merge(tx, quality, by = "city")

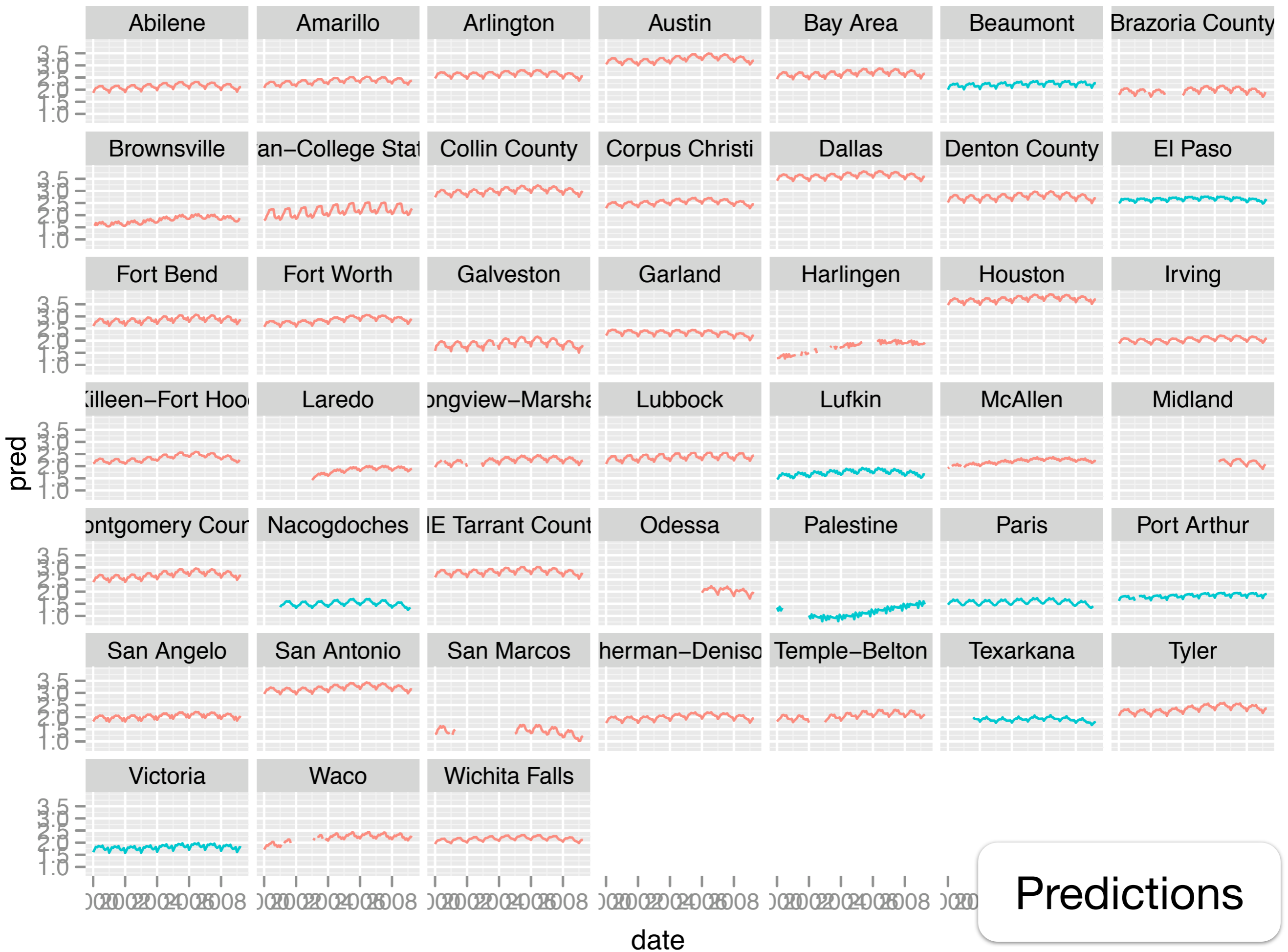
mfit <- lapply(models3, function(mod) {
  data.frame(
    resid = resid(mod),
    pred = predict(mod))
})
tx2 <- cbind(tx2, mfit[, -1])
```

Can you think of any potential problems with this line?

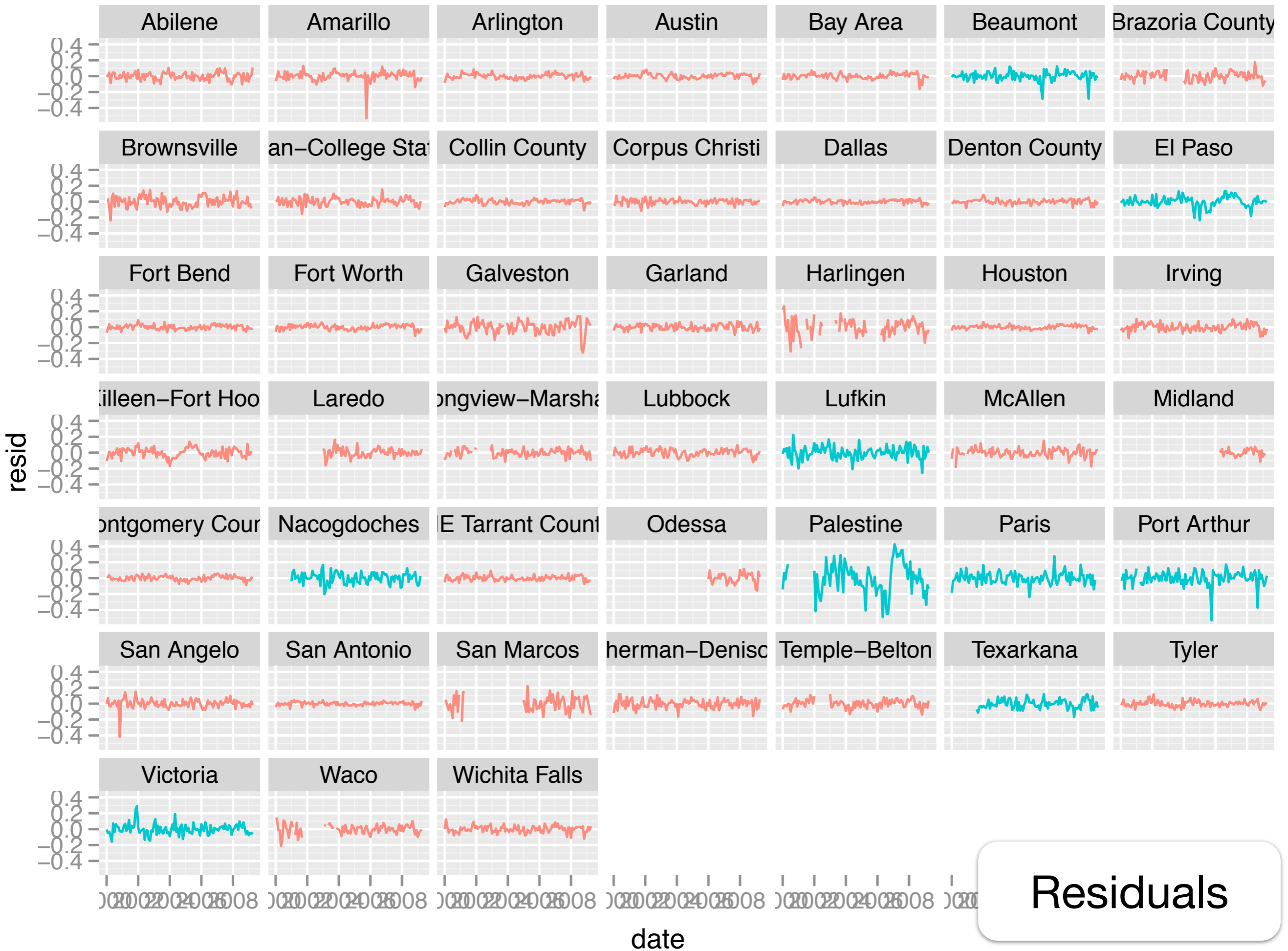




Raw data



Predictions



Residuals

# Conclusions

Simple (and relatively small) example, but shows how collections of models can be useful for gaining understanding.

Each attempt illustrated something new about the data.

Plyr made it easy to create and summarise collection of models, so we didn't have to worry about the mechanics.