Texas housing

Hadley Wickham

October 2009

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

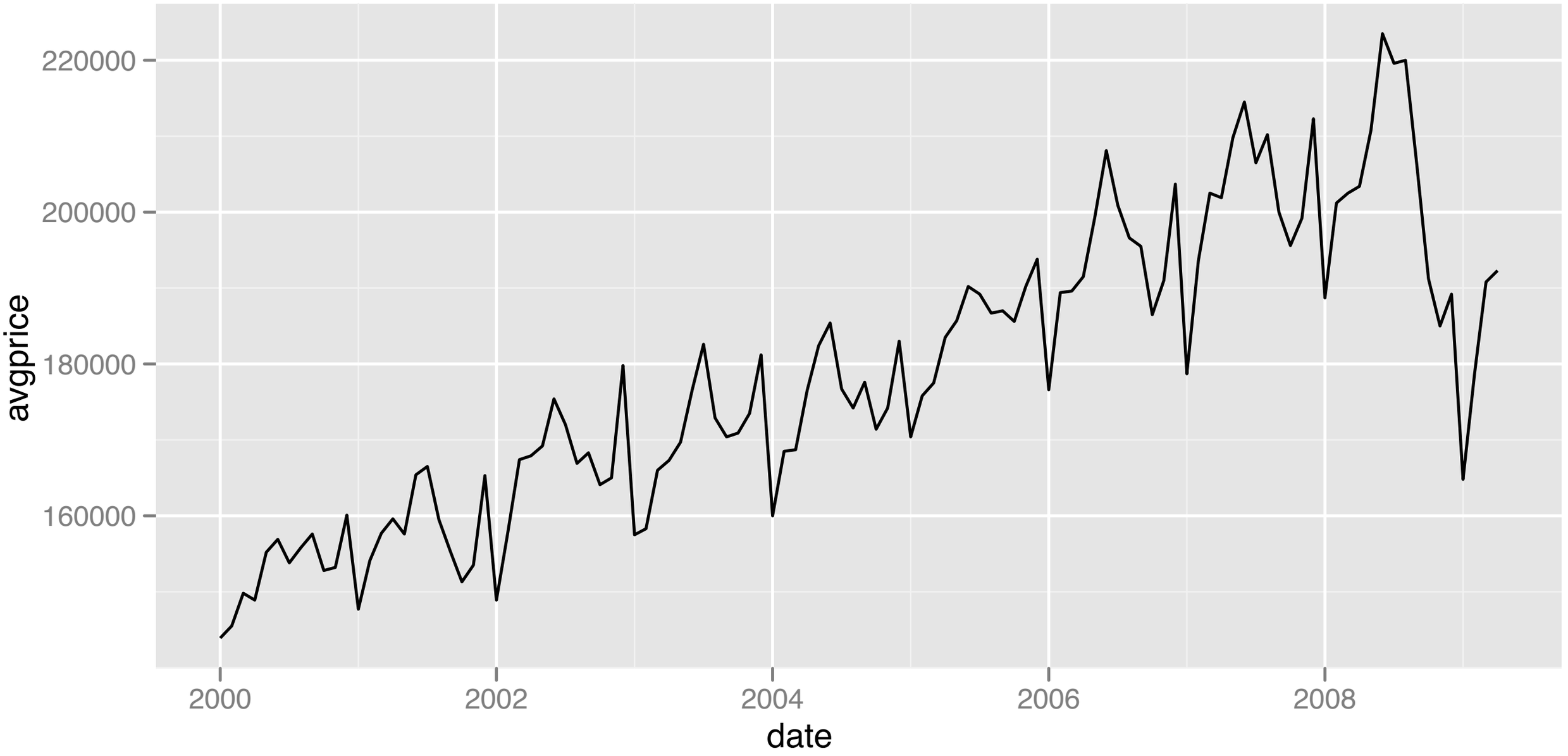
Average time on market

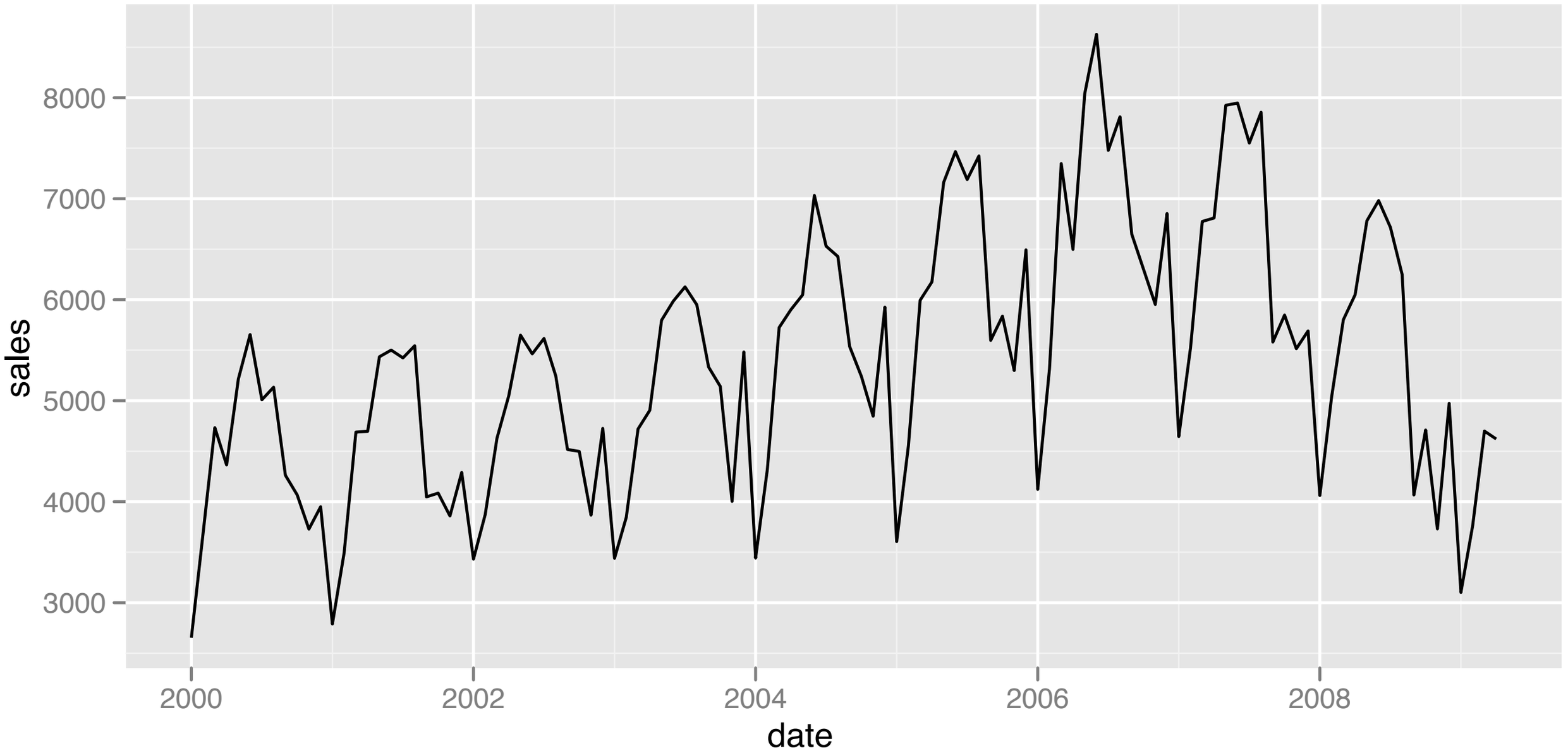
Strategy

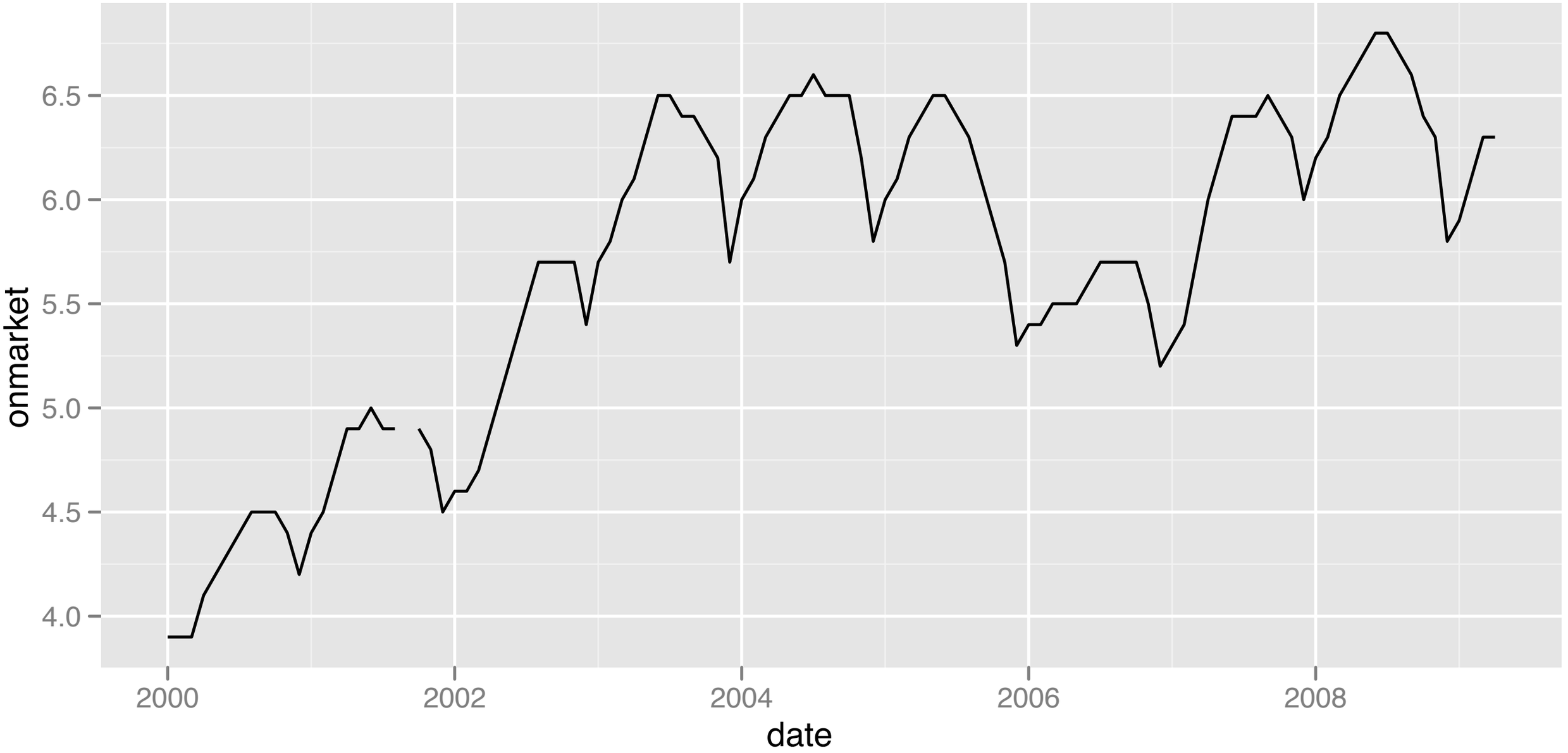
Start with a single city (Houston).

Explore patterns & fit models.

Apply models to all cities.


220000
200000
180000
160000
avgprice
2000
2002
2004
2006
2008
date

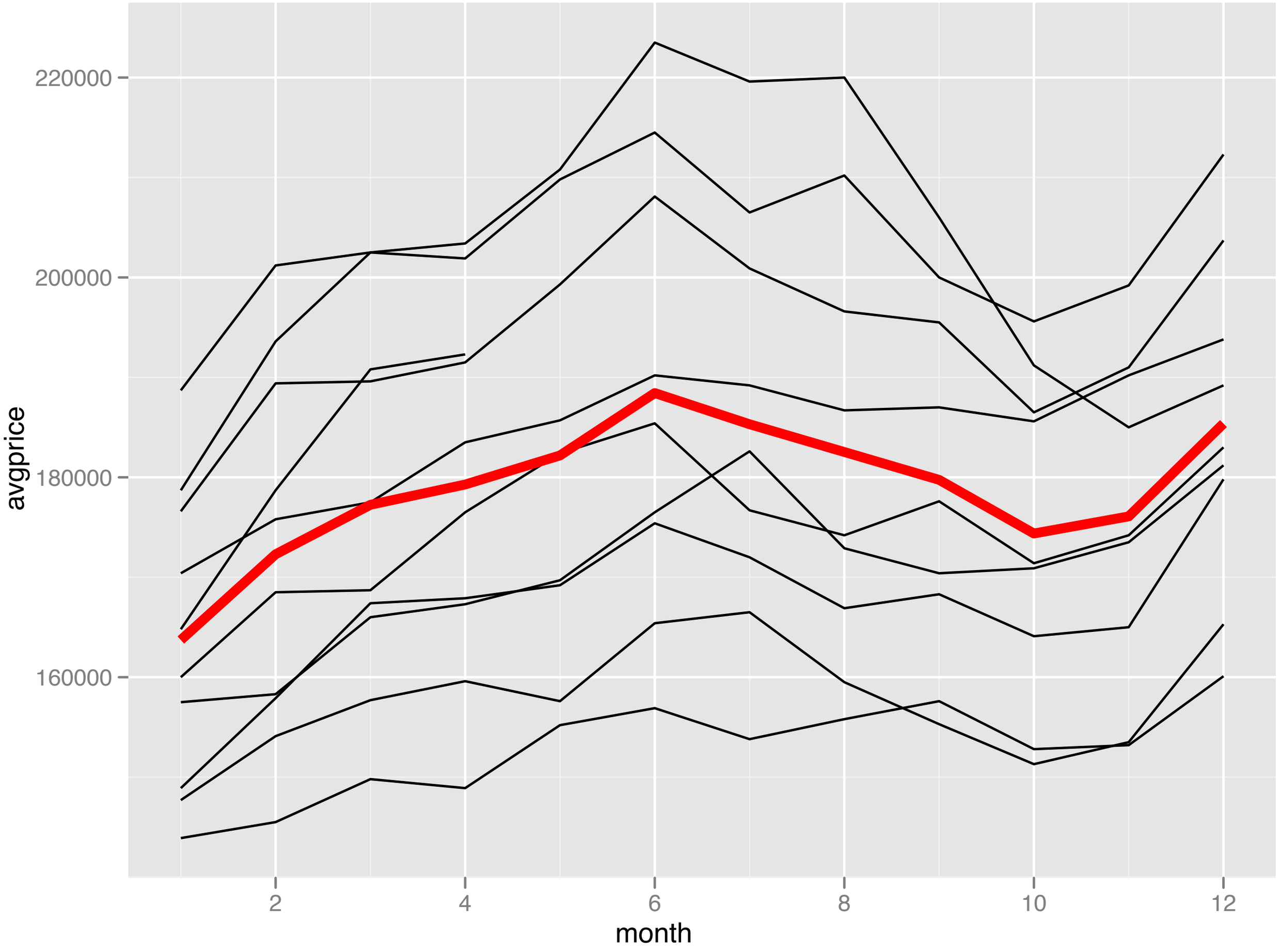

8000
7000
6000
5000
4000
3000
sales
2000
2002
2004
2006
2008
date

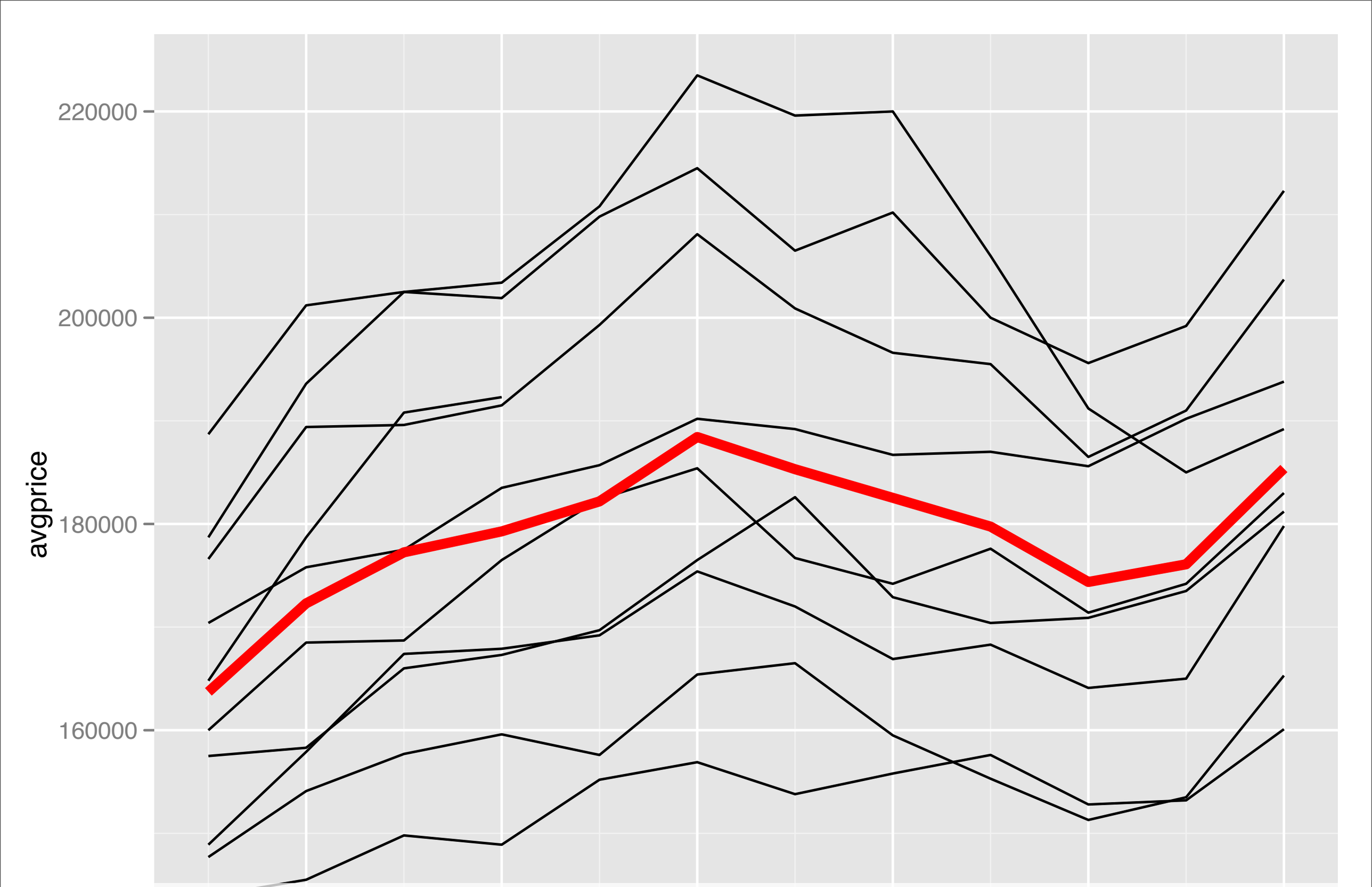

6.5
6.0
5.5
5.0
4.5
4.0
onmarket
2000
2002
2004
2006
2008
date

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?)


220000
200000
180000
160000
avgprice
2
4
6
8
10
12
month




```
qplot(month, avgprice, data = houston, geom = "line", group = year) +
  stat_summary(aes(group = 1), fun.y = "mean", geom = "line",
  colour = "red", size = 2, na.rm = TRUE)
```

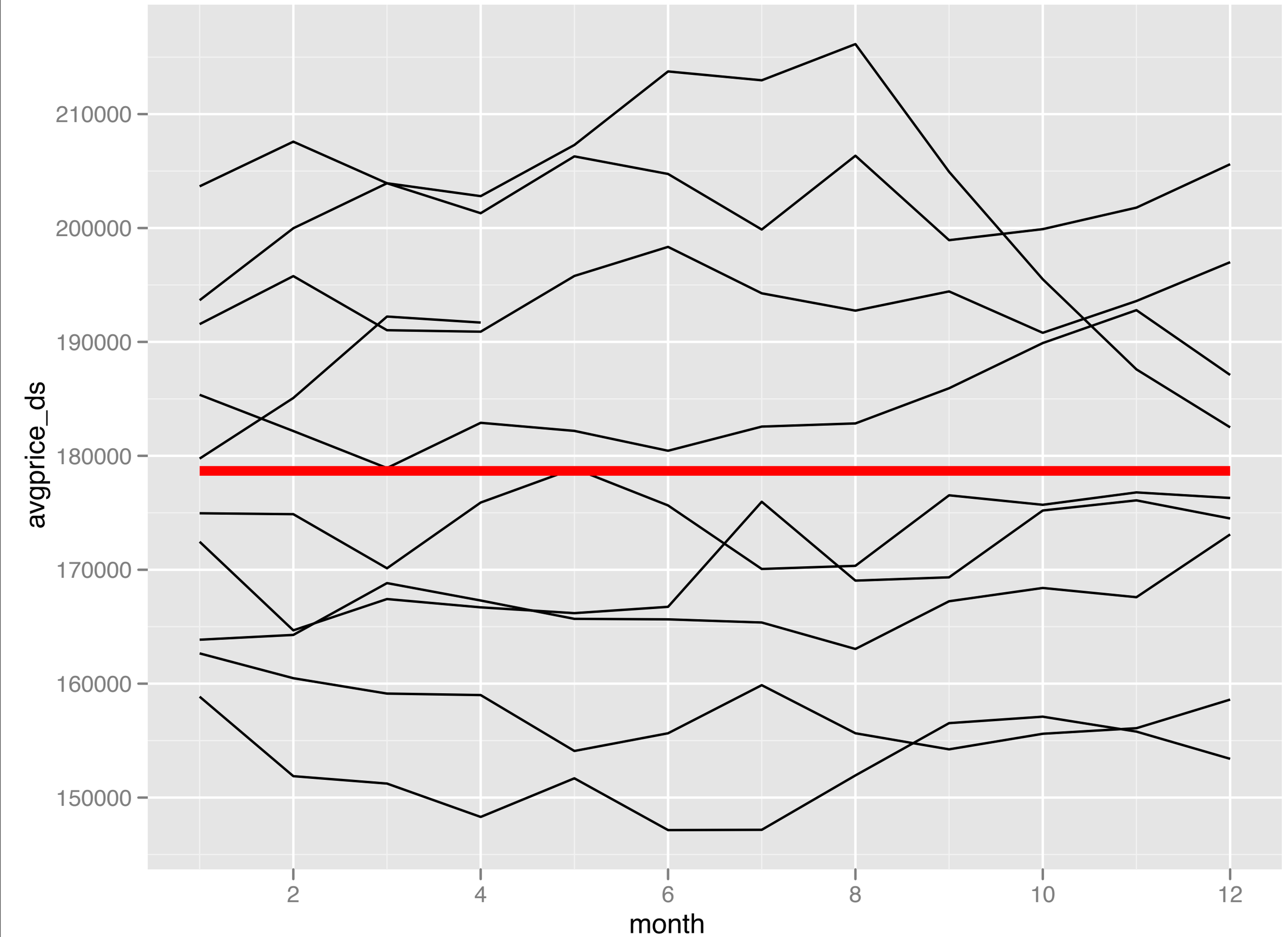
Challenge

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?

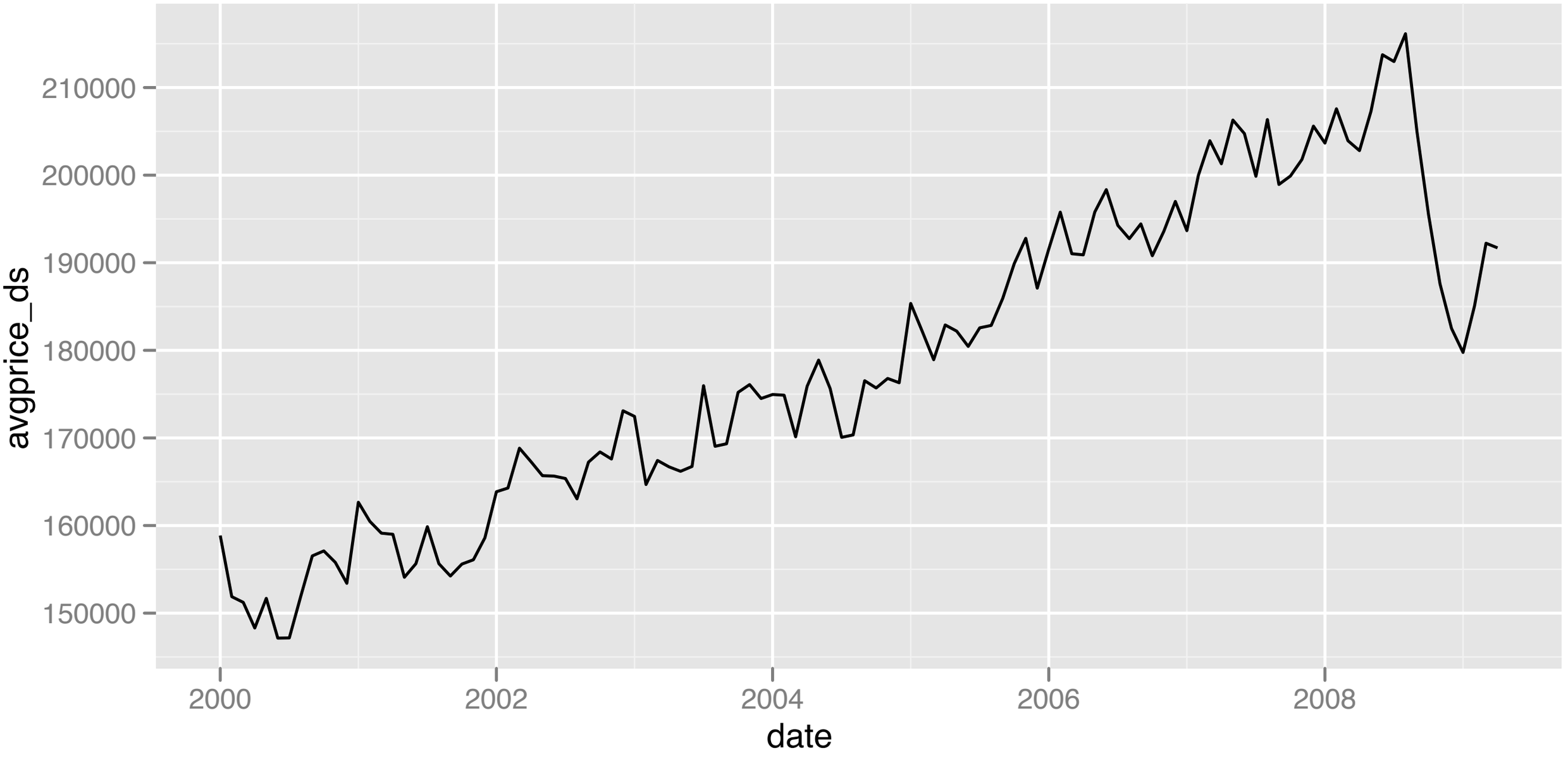
```
houston <- transform(houston,
  avgprice_ds = deseas(avgprice, month),
  listings_ds = deseas(listings, month),
  sales_ds = deseas(sales, month),
  onmarket_ds = deseas(onmarket, month)
)

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

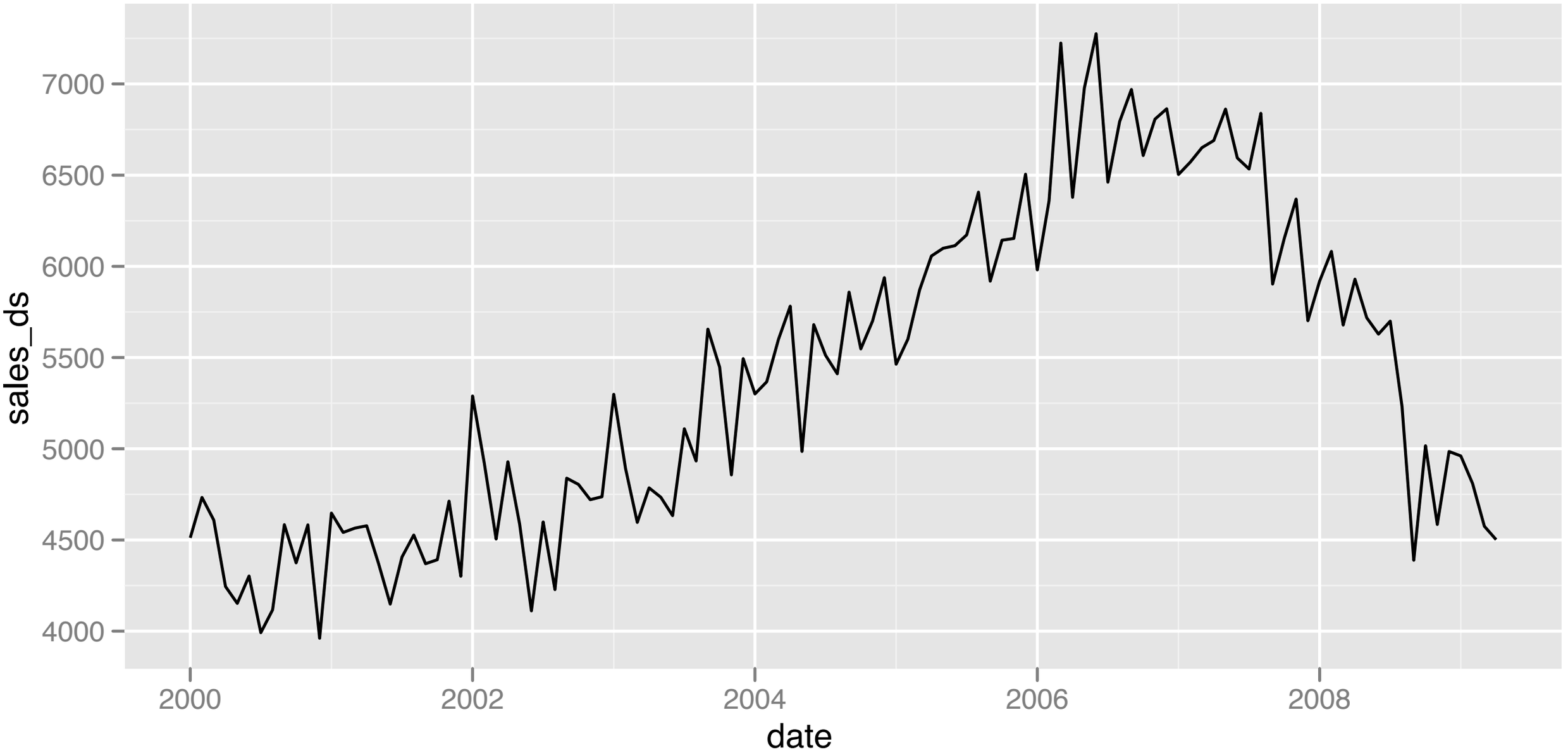

avgprice_ds
210000
200000
190000
180000
170000
160000
150000
2
4
6
8
10
12
month

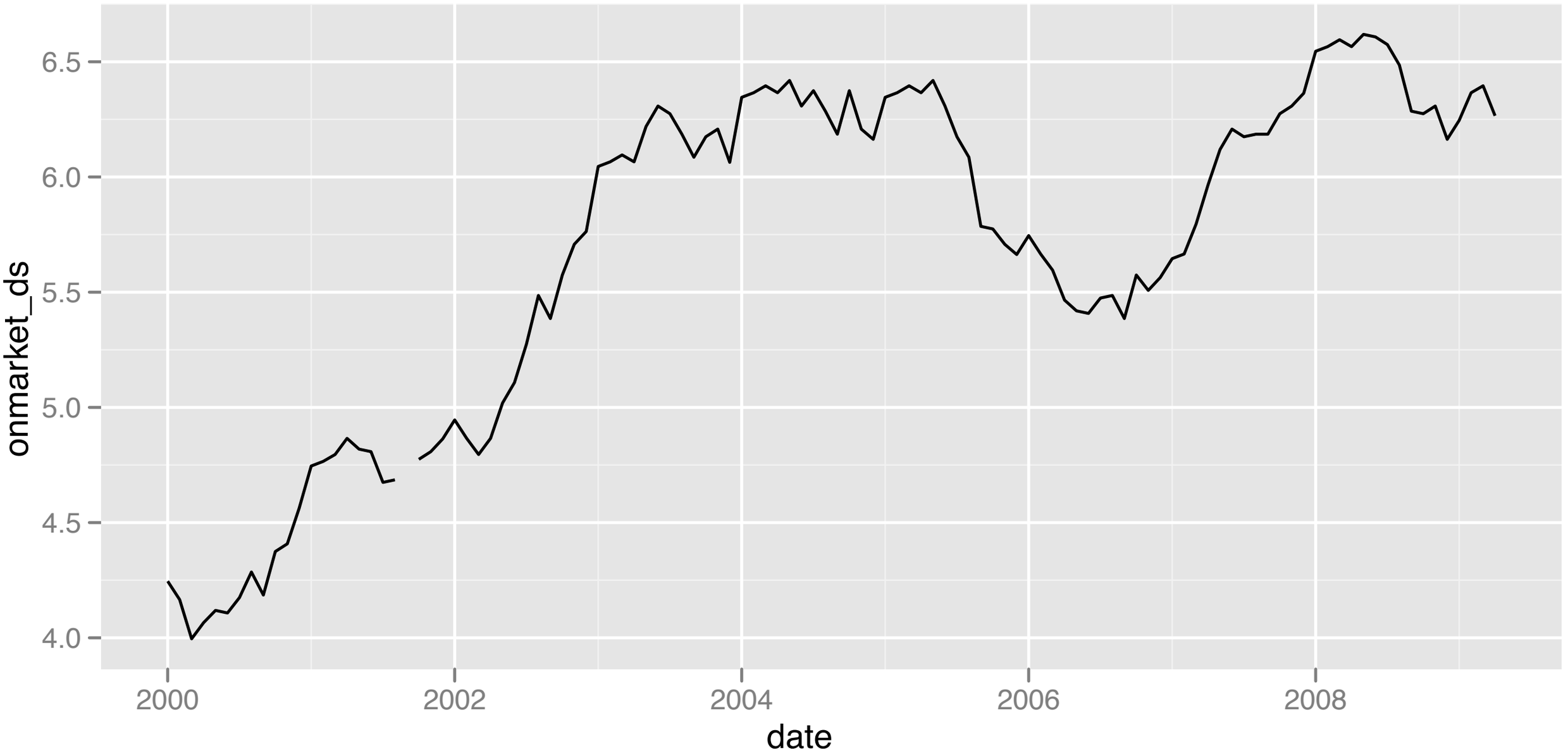
Model as tools

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.


avgprice_ds
210000
200000
190000
180000
170000
160000
150000
2000
2002
2004
2006
2008
date

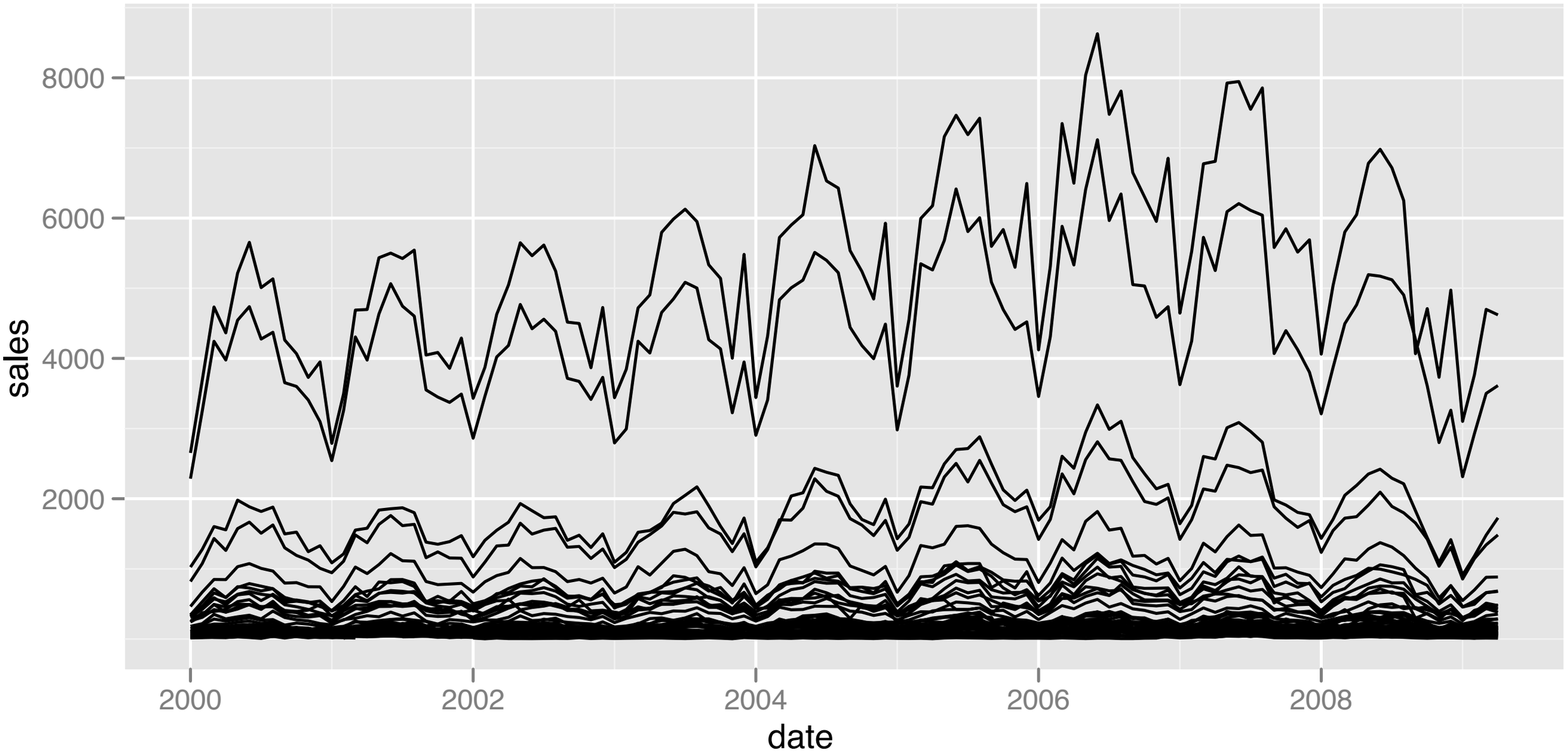

7000
6500
6000
5500
5000
4500
4000
sales_ds
2000
2002
2004
2006
2008
date


6.5
6.0
5.5
5.0
4.5
4.0
onmark_ds
2000
2002
2004
2006
2008
date

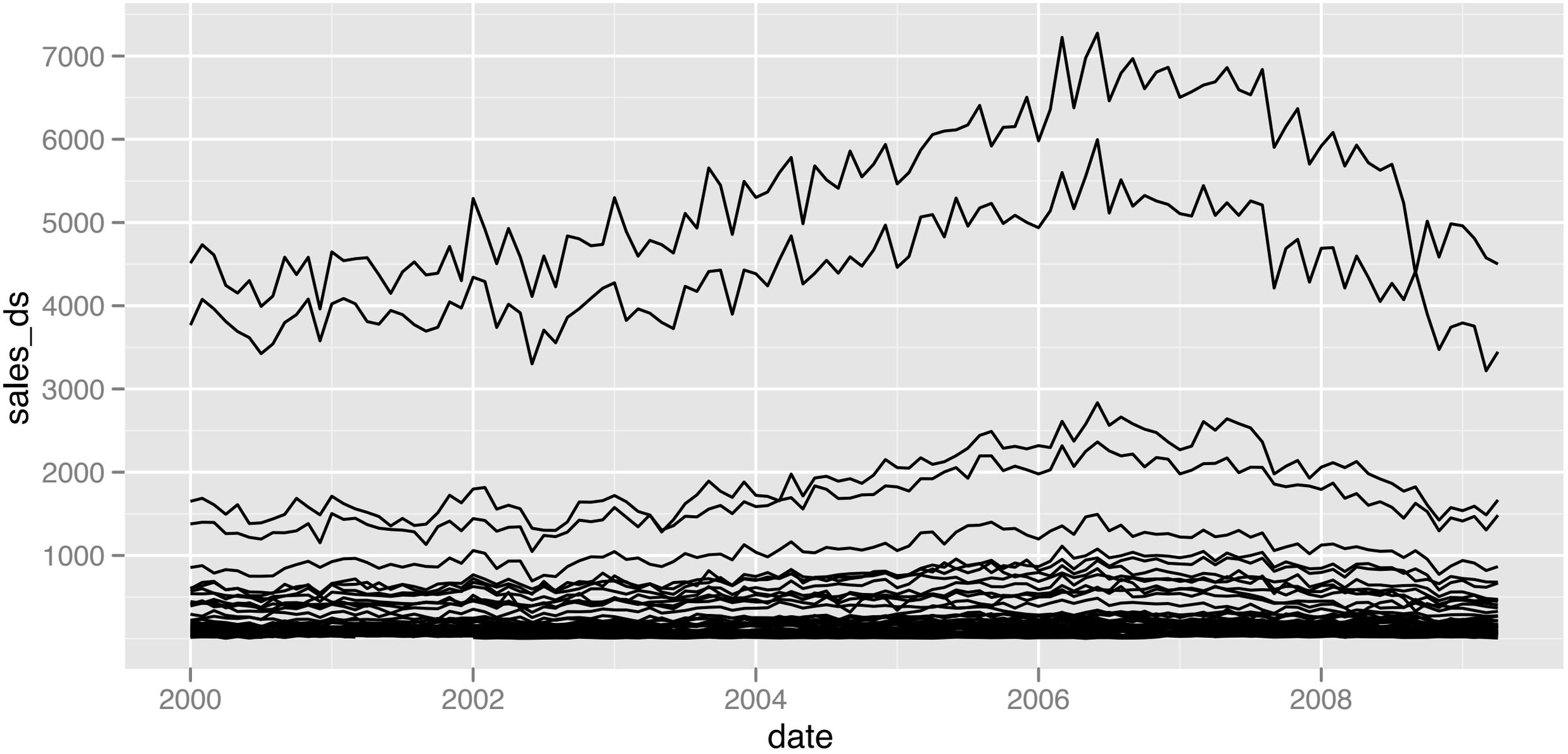
Summary

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.


8000
6000
4000
2000
sales
2000
2002
2004
2006
2008
date

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

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


7000
6000
5000
4000
3000
2000
1000
sales_ds
2000
2002
2004
2006
2008
date

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

`dlply`: 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

`dlply + ldply = ddply`

```
models <- dlply(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?

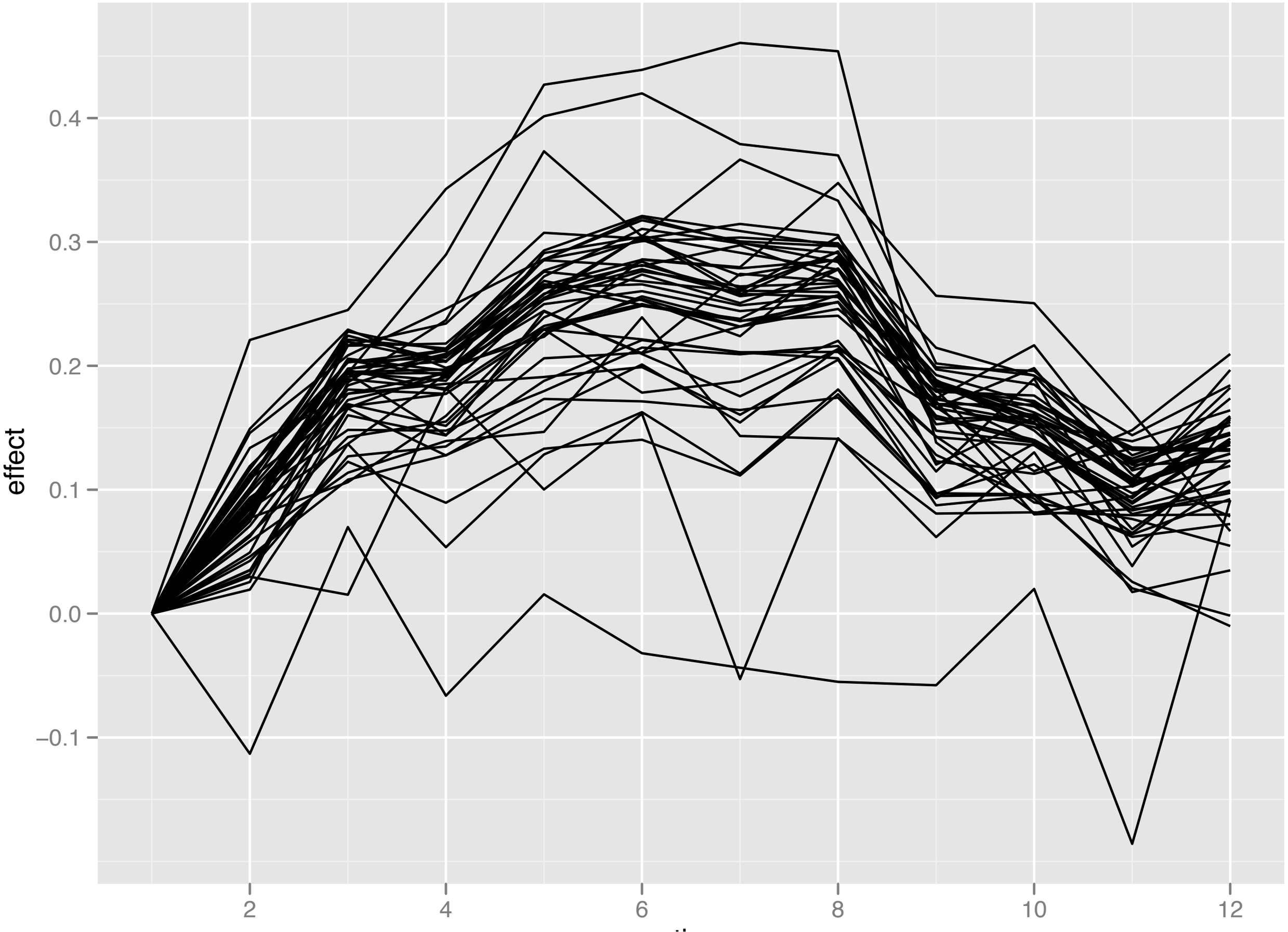
Turn to the person next to you and discuss for 2 minutes.

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

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

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

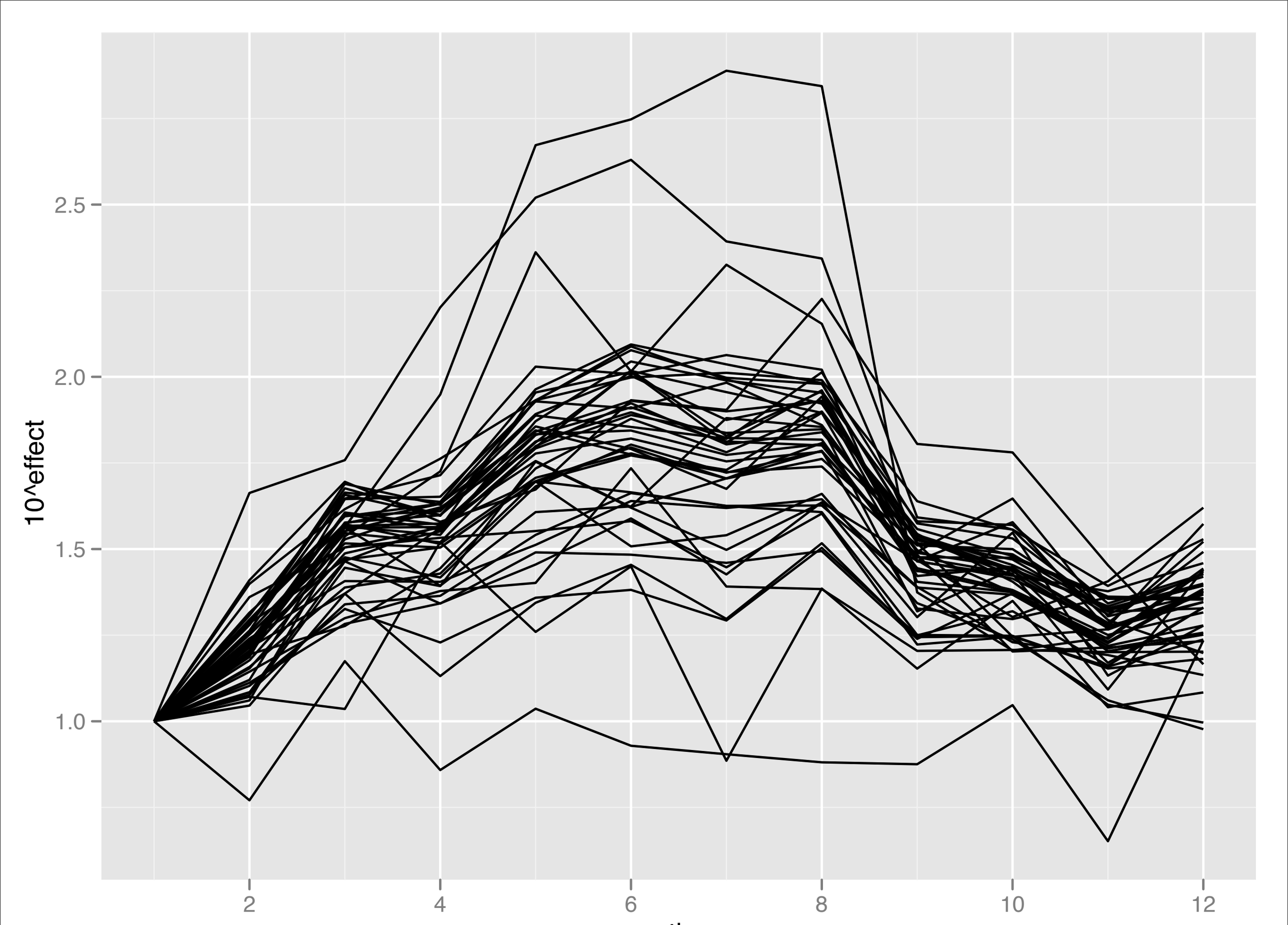
```
qplot(date, log10(sales), data = tx, geom = "line",
group = city)

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

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

Log transform sales to make coefficients comparable (ratios)

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")
```

Abilene | Amarillo | Arlington | Austin | Bay Area | Beaumont | Brazoria County

Brownsville | an–College Stat | Collin County | Corpus Christi | Dallas | Denton County | El Paso

Fort Bend | Fort Worth | Galveston | Garland | Harlingen | Houston | Irving

Killeen–Fort Hood | Laredo | ongview–Marsha | Lubbock | Lufkin | McAllen | Midland

ontgomery Coun | Nacogdoches | IE Tarrant Count | Odessa | Palestine | Paris | Port Arthur

San Angelo | San Antonio | San Marcos | herman–Deniso | Temple–Belton | Texarkana | Tyler

Victoria | Waco | Wichita Falls

10^effect

2.5
2.0
1.5
1.0

2 4 6 8 10 12

month

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

What should we do next?

What do you think?

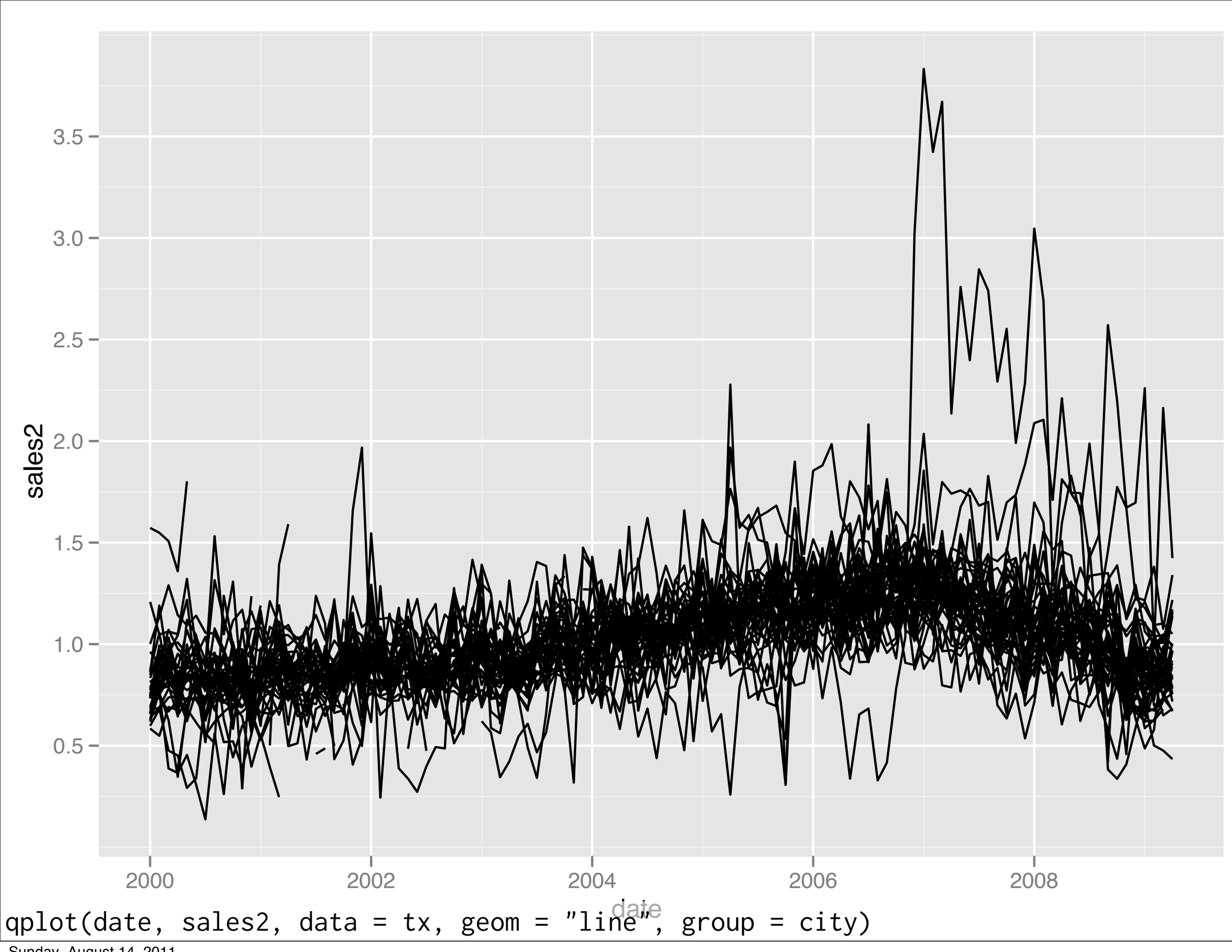
You have 30 seconds to come up with (at least) one idea.

My ideas

Fit a single model, `log(sales) ~ city * factor(month)`, and look at residuals

Fit individual models, `log(sales) ~ factor(month) + ns(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)
```

Abilene | Amarillo | Arlington | Austin | Bay Area | Beaumont | Brazoria County

Brownsville | an–College Stat | Collin County | Corpus Christi | Dallas | Denton County | El Paso

Fort Bend | Fort Worth | Galveston | Garland | Harlingen | Houston | Irving

illeen–Fort Hoo | Laredo | ongview–Marsha | Lubbock | Lufkin | McAllen | Midland

ontgomery Cour | Nacogdoches | IE Tarrant Count | Odessa | Palestine | Paris | Port Arthur

San Angelo | San Antonio | San Marcos | herman–Deniso | Temple–Belton | Texarkana | Tyler

Victoria | Waco | Wichita Falls

sales2

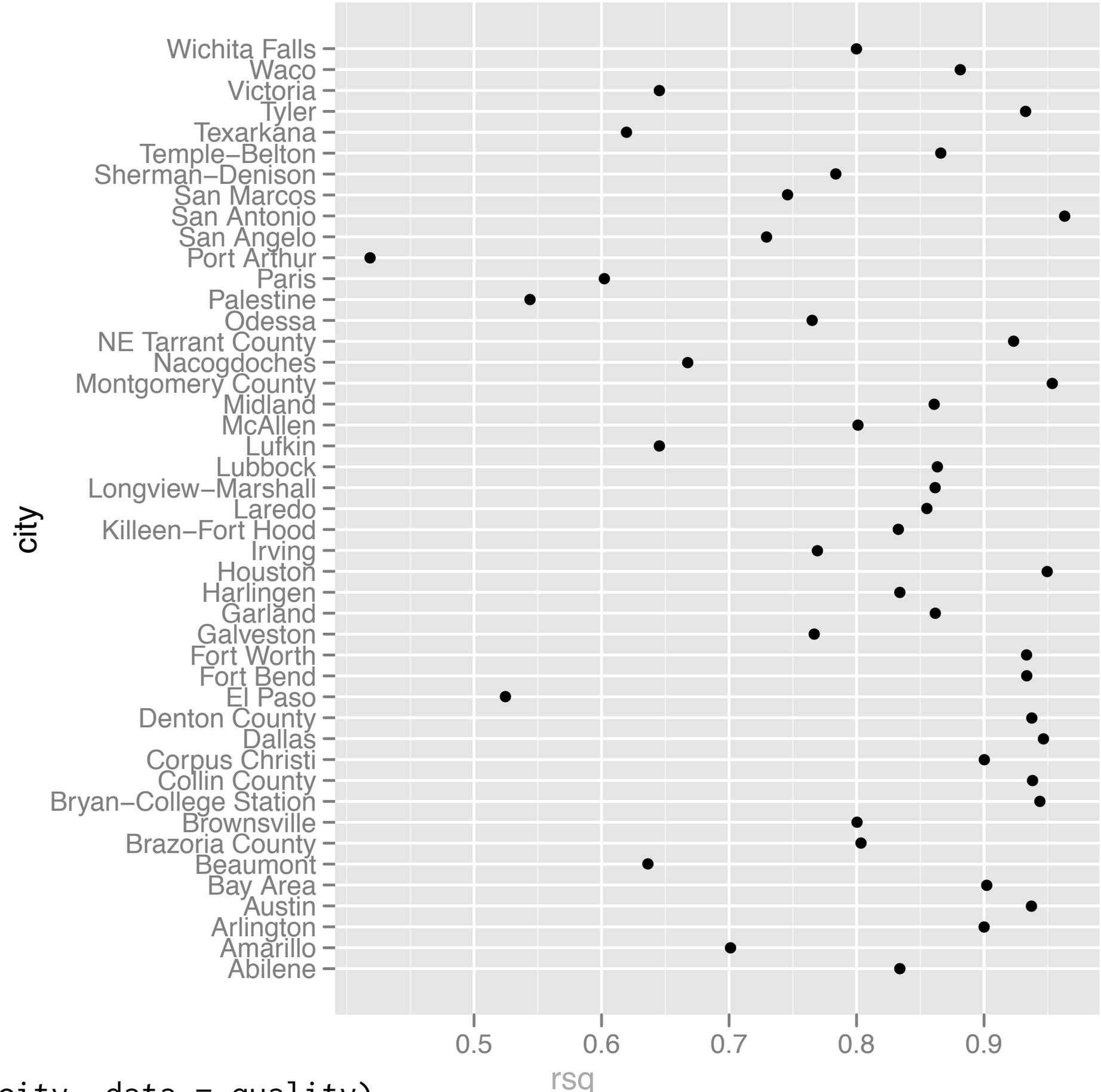
0.5 1.0 1.5 2.0 2.5 3.0 3.5

2000 02 04 06 08

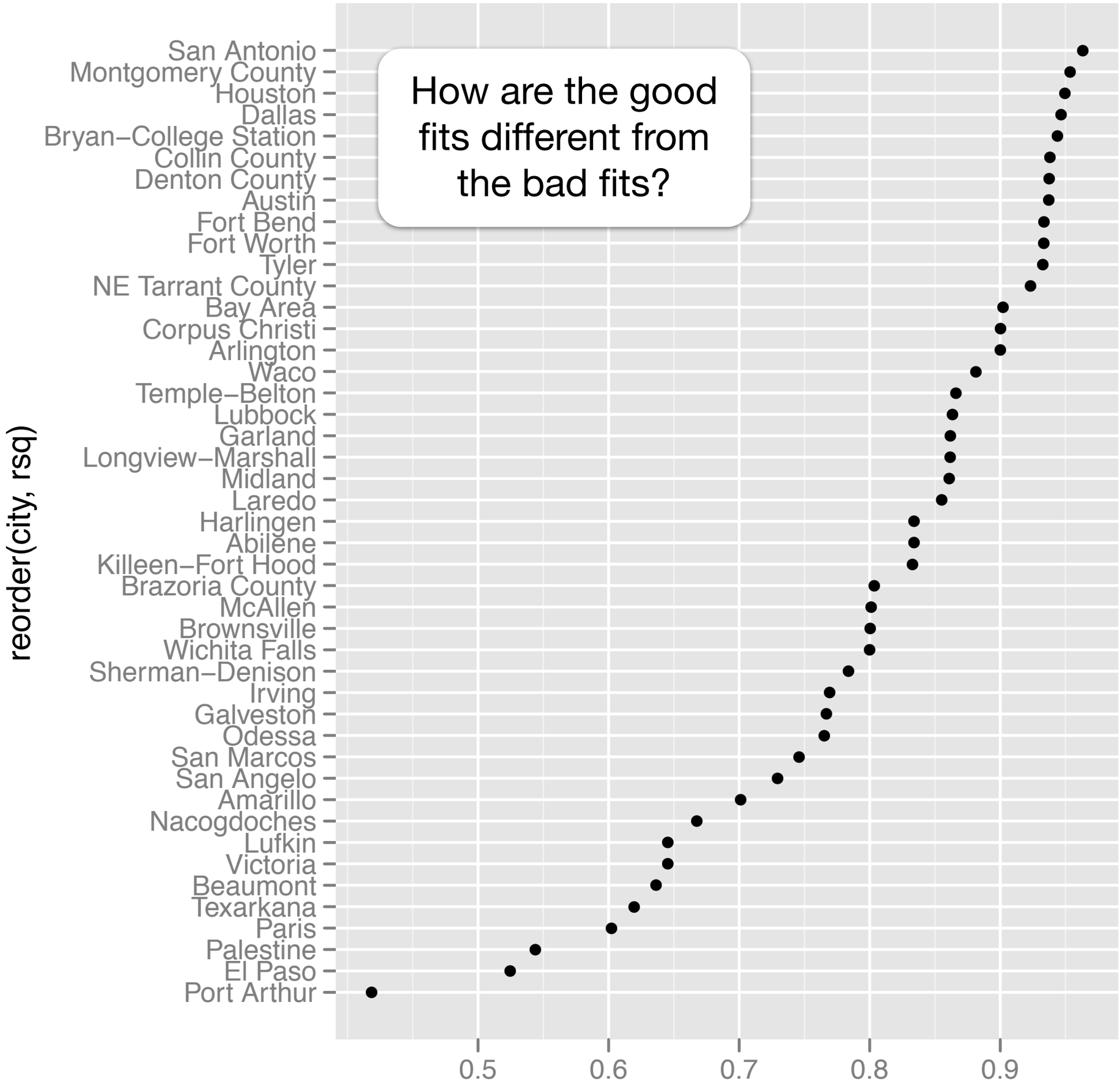
date

last_plot() + facet_wrap(~ city)

```r
# 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 <- dlply(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")

cities <- dlply(tx, "city")

mfit <- mdply(cbind(mod = models3, df = cities),
  function(mod, df) {
    data.frame(
      city = df$city, date = df$date,
      resid = resid(mod), pred = predict(mod))
})
tx2 <- merge(tx2, mfit)
```

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

cities <- dlply(tx, "city")

mfit <- mdply(cbind(mod = models3, df = cities),
  function(mod, df) {
    data.frame(
      city = df$city, date = df$date,
      resid = resid(mod), pred = predict(mod))
})
tx2 <- merge(tx2, mfit)
```

Takes each row of input and feeds it to function

Abilene | Amarillo | Arlington | Austin | Bay Area | Beaumont | Brazoria County

Brownsville | an–College Stat | Collin County | Corpus Christi | Dallas | Denton County | El Paso

Fort Bend | Fort Worth | Galveston | Garland | Harlingen | Houston | Irving

illeen–Fort Hoo | Laredo | ongview–Marsha | Lubbock | Lufkin | McAllen | Midland

ontgomery Cour | Nacogdoches | IE Tarrant Count | Odessa | Palestine | Paris | Port Arthur

San Angelo | San Antonio | San Marcos | herman–Deniso | Temple–Belton | Texarkana | Tyler

Victoria | Waco | Wichita Falls

log10(sales)

date

Raw data

Abilene | Amarillo | Arlington | Austin | Bay Area | Beaumont | Brazoria County

Brownsville | an–College Stat | Collin County | Corpus Christi | Dallas | Denton County | El Paso

Fort Bend | Fort Worth | Galveston | Garland | Harlingen | Houston | Irving

illeen–Fort Hoo | Laredo | ongview–Marsha | Lubbock | Lufkin | McAllen | Midland

ontgomery Cour | Nacogdoches | IE Tarrant Count | Odessa | Palestine | Paris | Port Arthur

San Angelo | San Antonio | San Marcos | herman–Deniso | Temple–Belton | Texarkana | Tyler

Victoria | Waco | Wichita Falls

pred

date

Predictions

Abilene | Amarillo | Arlington | Austin | Bay Area | Beaumont | Brazoria County
Brownsville | an–College Stat | Collin County | Corpus Christi | Dallas | Denton County | El Paso
Fort Bend | Fort Worth | Galveston | Garland | Harlingen | Houston | Irving
illeen–Fort Hoo | Laredo | ongview–Marsha | Lubbock | Lufkin | McAllen | Midland
ontgomery Cour | Nacogdoches | E Tarrant Count | Odessa | Palestine | Paris | Port Arthur
San Angelo | San Antonio | San Marcos | herman–Denisc | Temple–Belton | Texarkana | Tyler
Victoria | Waco | Wichita Falls

0.4 0.2 0.0 −0.2 −0.4

resid

date

Residuals

Your turn

Pick one of the other variables and perform a similar exploration, working through the same steps.

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.

Sunday, August 14, 2011