Time and space

Hadley Wickham

October 2009

1. New data: baby names by state
2. Visualise time (done!)
3. Visualise time conditional on space
4. Visualise space
5. Visualise space conditional on time

Baby names by state

Top 100 male and female baby names for each state, 1960–2008.

480,000 records (100 * 50 * 2 * 48)

Slightly different variables: state, year, name, sex and **number**.

To keep the data manageable, we will look at the top 25 names of all time.

CC BY http://www.flickr.com/photos/the_light_show/2586781132

Subset

Easier to compare states if we have proportions. To calculate proportions, need births. But could only find data from 1981.

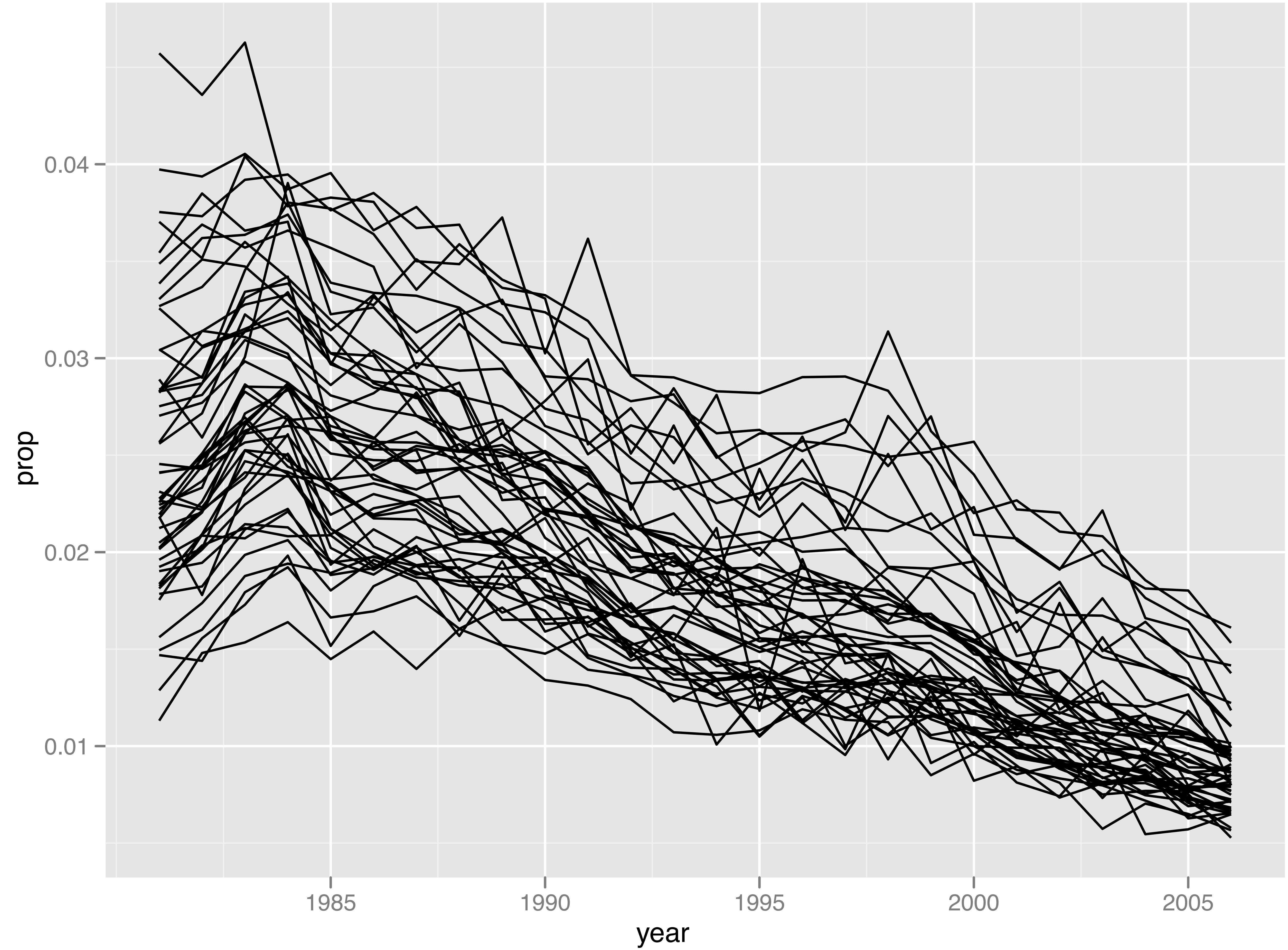
Then selected 30 names that occurred fairly frequently, and had interesting patterns.

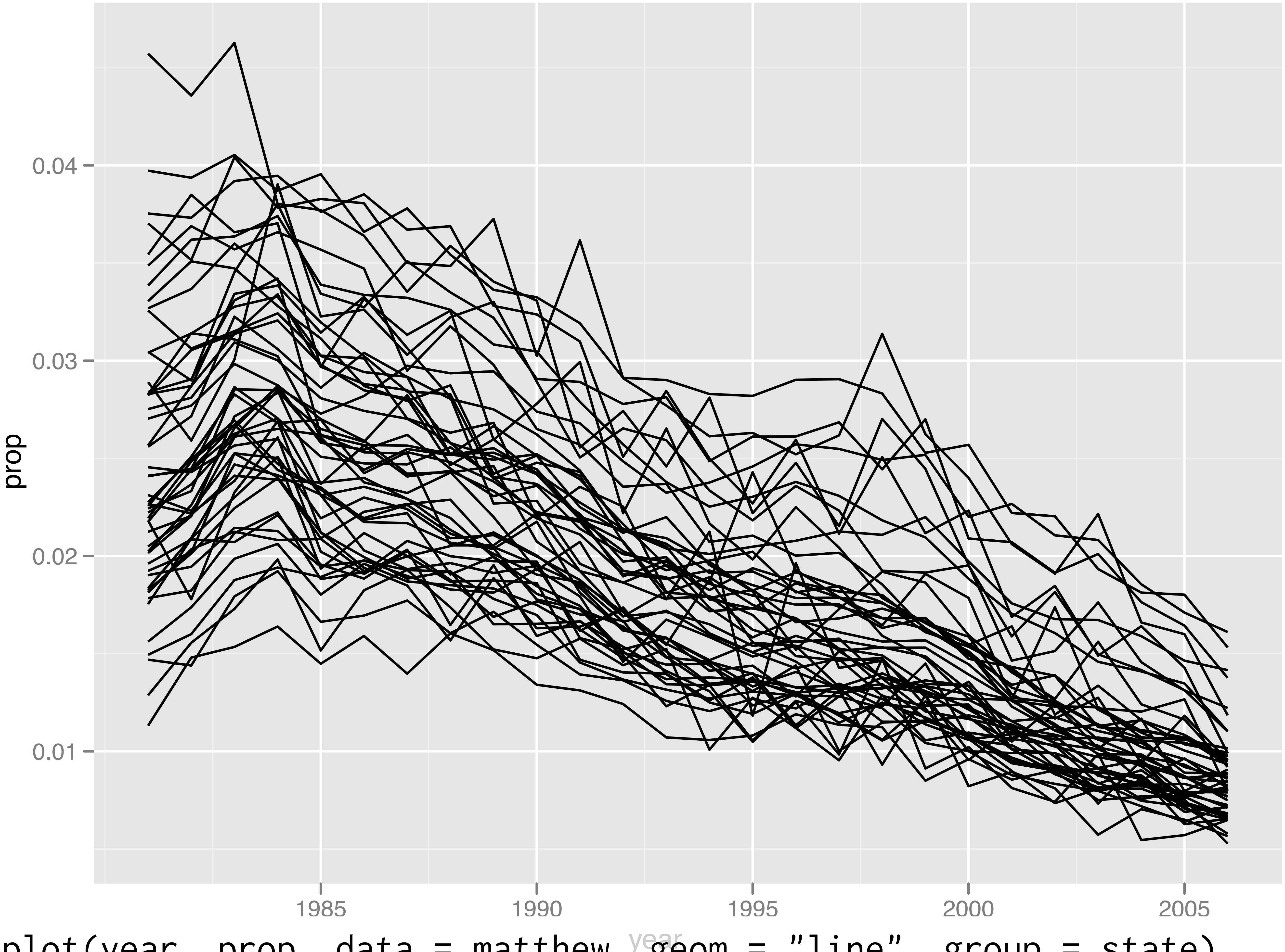
Aaron Alex Allison Alyssa Angela Ashley
Carlos Chelsea Christian Eric Evan
Gabriel Jacob Jared Jennifer Jonathan
Juan Katherine Kelsey Kevin Matthew
Michelle Natalie Nicholas Noah Rebecca
Sara Sarah Taylor Thomas

Getting started

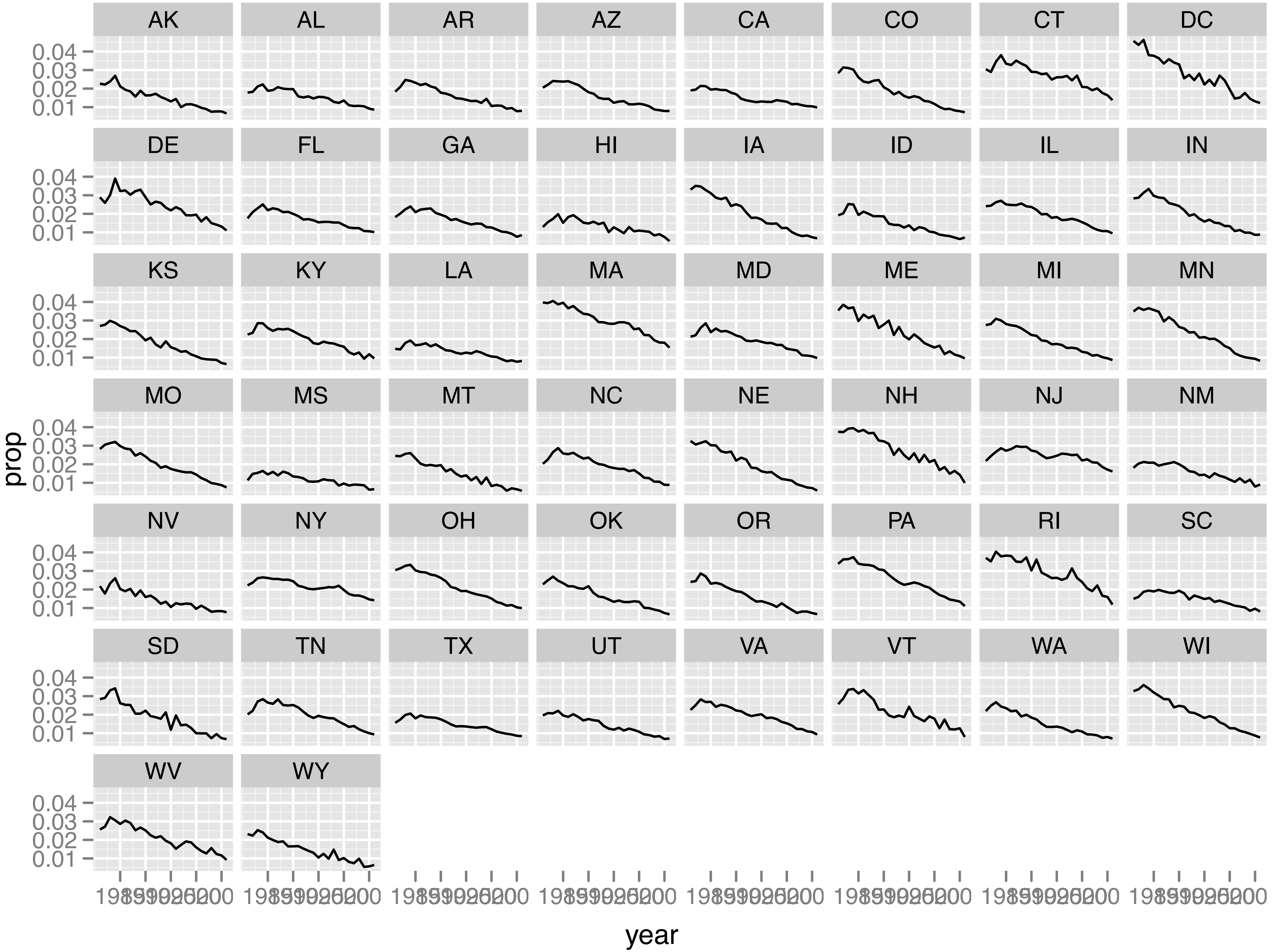
```
library(ggplot2)
library(plyr)

bnames <- read.csv(”interesting-names.csv”,
  stringsAsFactors = F)

matthew <- subset(bnames, name == ”Matthew”)
```

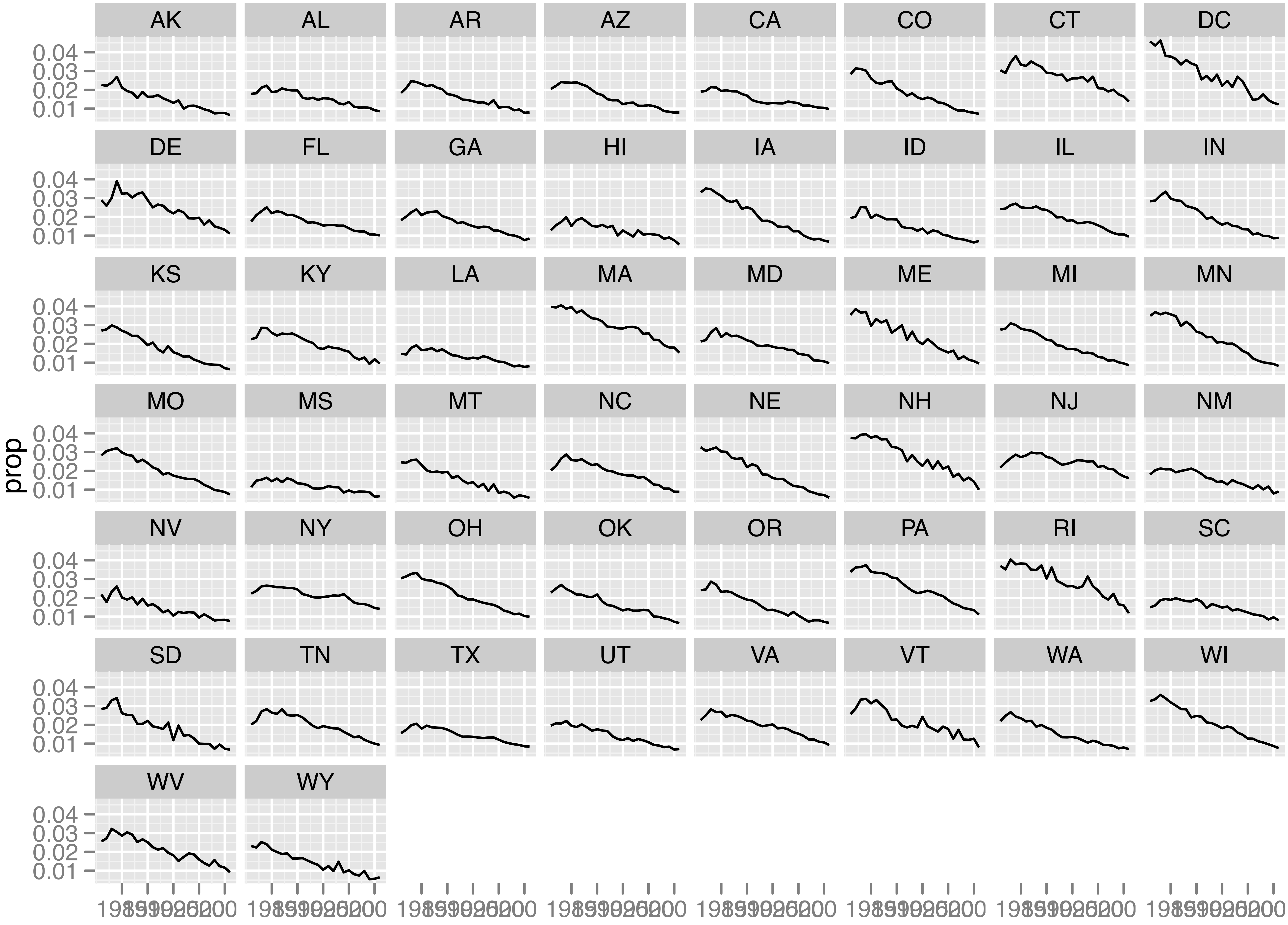
Time | Space


0.04
0.03
0.02
0.01
prop
1985
1990
1995
2000
2005
year



```
qplot(year, prop, data = matthew, geom = "line", group = state)
```


AK
AL
AR
AZ
CA
CO
CT
DC
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY
0.04
0.03
0.02
0.01
prop
year



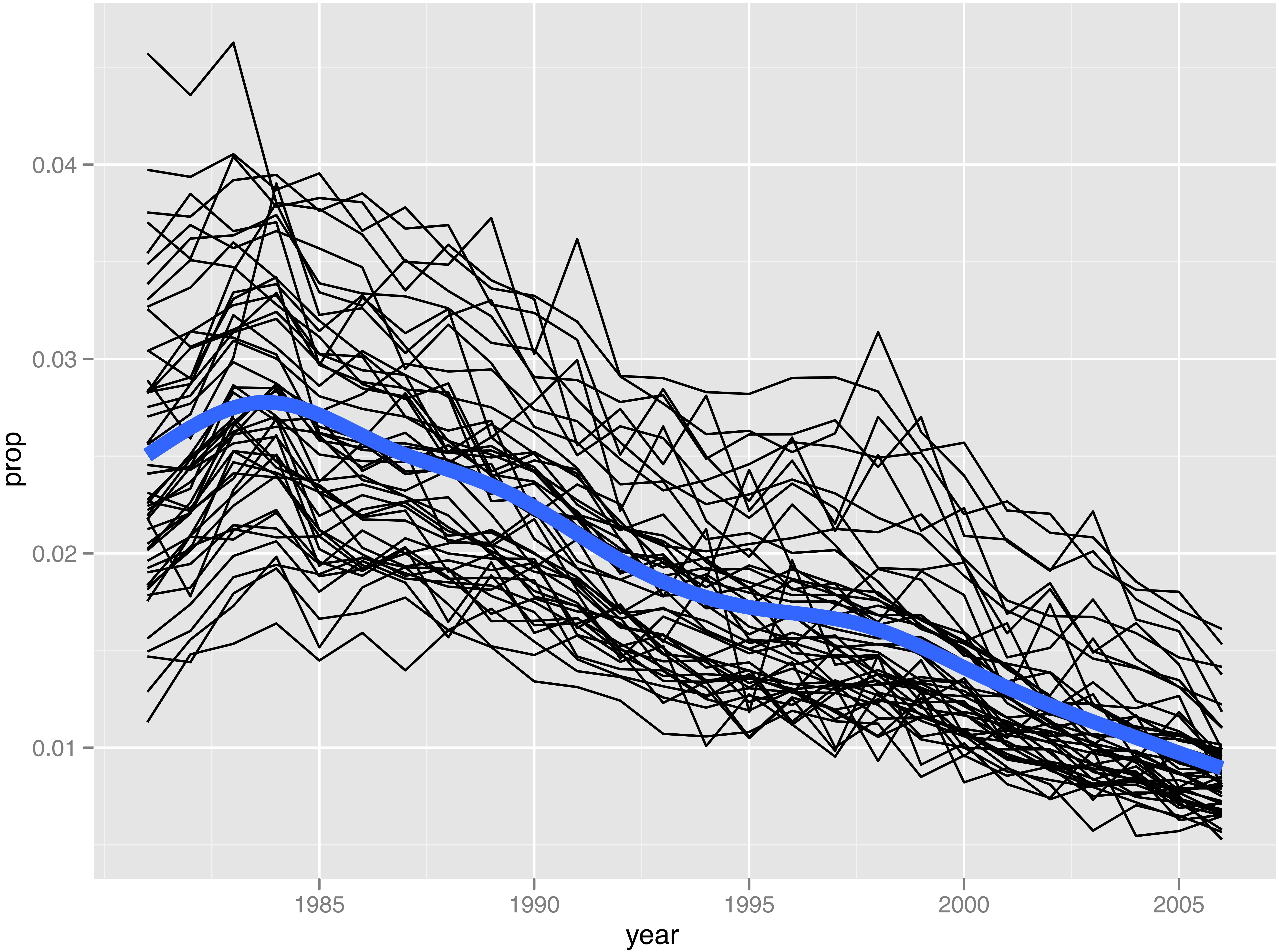
```
last_plot() + facet_wrap(~ state)
```

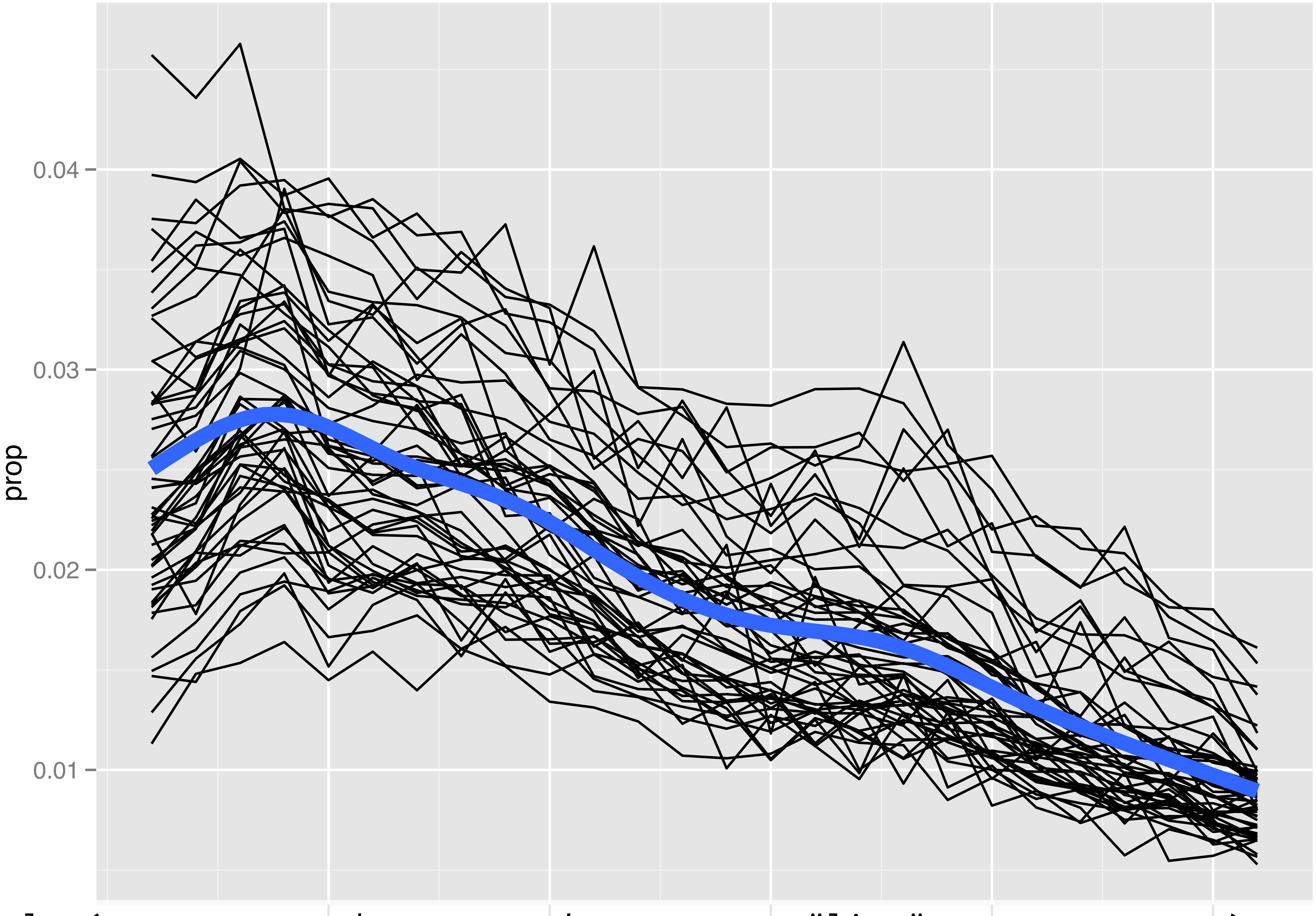
Your turn

Pick some names out of the list and explore. What do you see?

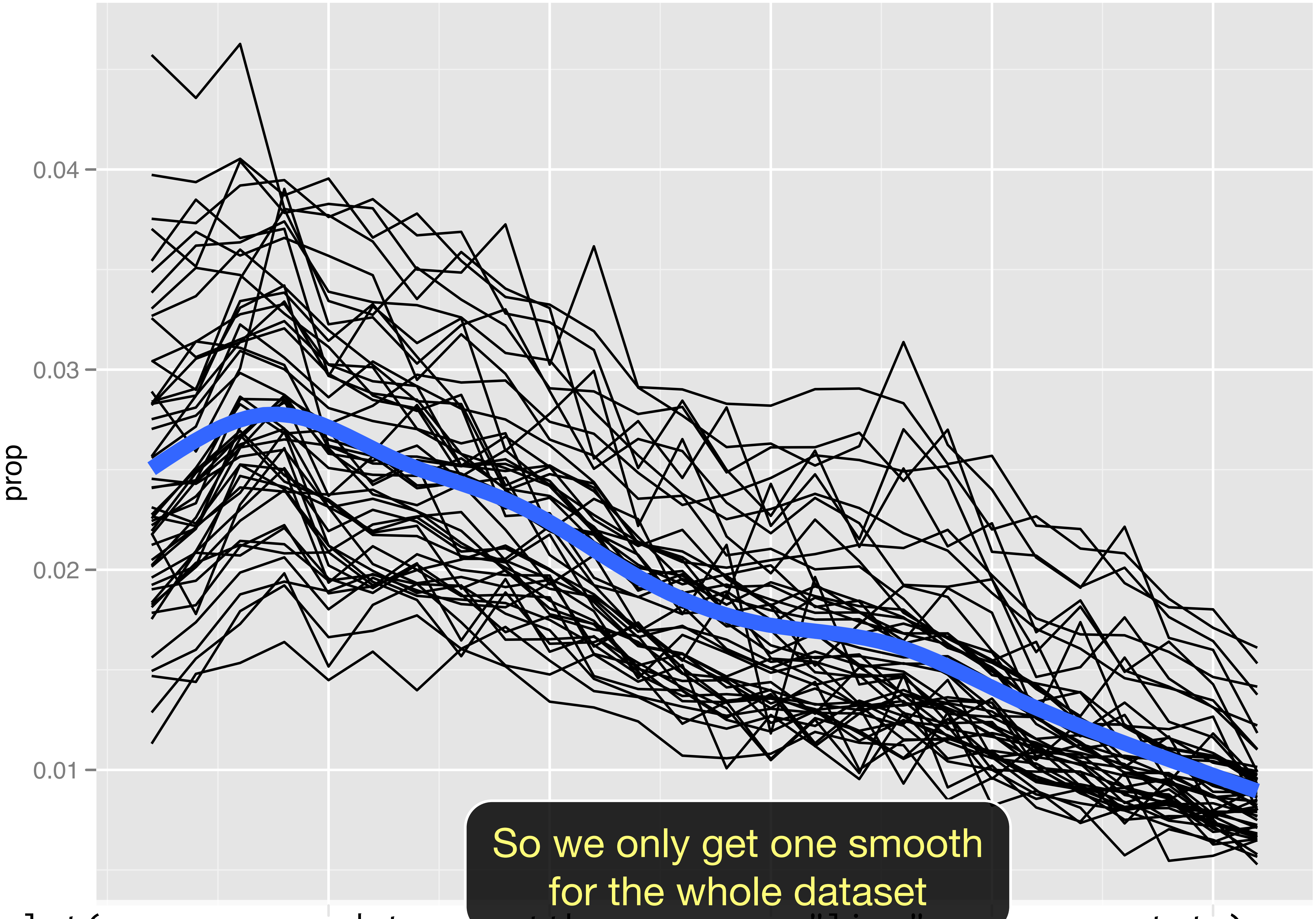
Can you write a function that plots the trend for a given name?

```
show_name <- function(name) {
  name <- bnames[bnames$name == name, ]
  qplot(year, prop, data = name, geom = ”line”,
    group = state)
}

show_name(”Jessica”)
show_name(”Aaron”)
show_name(”Juan”) + facet_wrap(~ state)
```


0.04
0.03
0.02
0.01
prop
1985
1990
1995
2000
2005
year



```
qplot(year, prop, data = matthew, geom = "line", group = state) +
  geom_smooth(aes(group = 1), se = F, size = 3)
```



So we only get one smooth for the whole dataset

```
qplot(year, prop, data = matthew, geom = "line", group = state) +
  geom_smooth(aes(group = 1), se = F, size = 3)
```

Two useful tools

Smoothing: can be easier to perceive overall trend by smoothing individual functions

Indexing: remove initial differences by indexing to the first value.

Not that useful here, but good tools to have in your toolbox.

```
library(mgcv)
smooth <- function(y, x) {
  as.numeric(predict(gam(y ~ s(x))))
}

matthew <- ddply(matthew, ”state”, transform,
  prop_s = smooth(prop, year))

qplot(year, prop_s, data = matthew, geom = ”line”,
  group = state)
```

```
index <- function(y, x) {
  y / y[order(x)[1]]
}

matthew <- ddply(matthew, ”state”, transform,
  prop_i = index(prop, year),
  prop_si = index(prop_s, year))

qplot(year, prop_i, data = matthew, geom = ”line”,
  group = state)
qplot(year, prop_si, data = matthew, geom = ”line”,
  group = state)
```

Your turn

Create a plot to show all names simultaneously. Does smoothing every name in every state make it easier to see patterns?

Hint: run the following R code on the next slide to eliminate names with less than 10 years of data

```
longterm <- ddply(bnames, c(”name”, ”state”),
function(df) {
  if (nrow(df) > 10) {
    df
  }
})
```

```
qplot(year, prop, data = bnames, geom = ”line”,
  group = state, alpha = I(1 / 4)) +
  facet_wrap(~ name)

longterm <- ddply(longterm, c(”name”, ”state”),
  transform, prop_s = smooth(prop, year))

qplot(year, prop_s, data = longterm, geom = ”line”,
  group = state, alpha = I(1 / 4)) +
  facet_wrap(~ name)
last_plot() + facet_wrap(scales = ”free_y”)
```

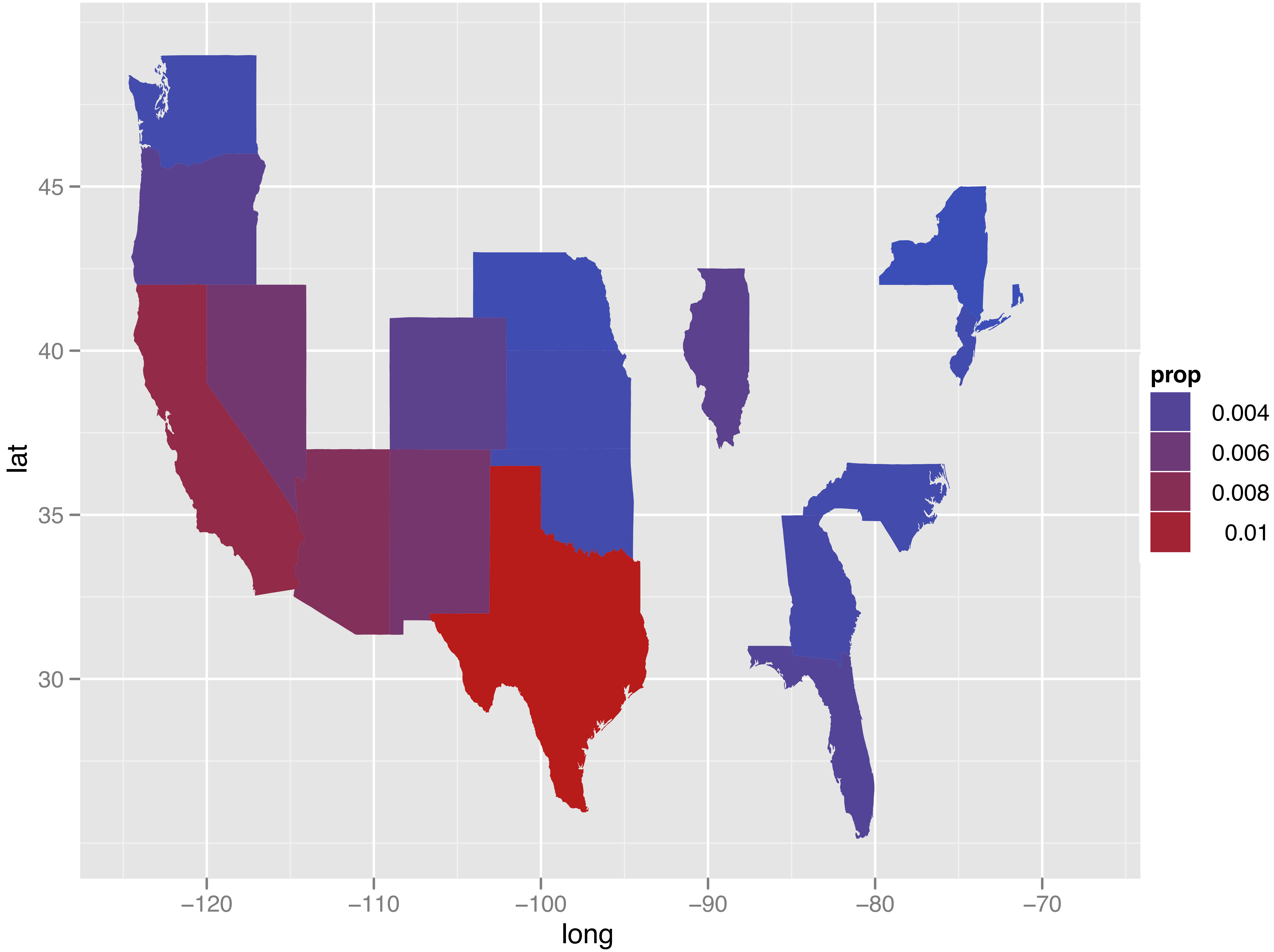
Space

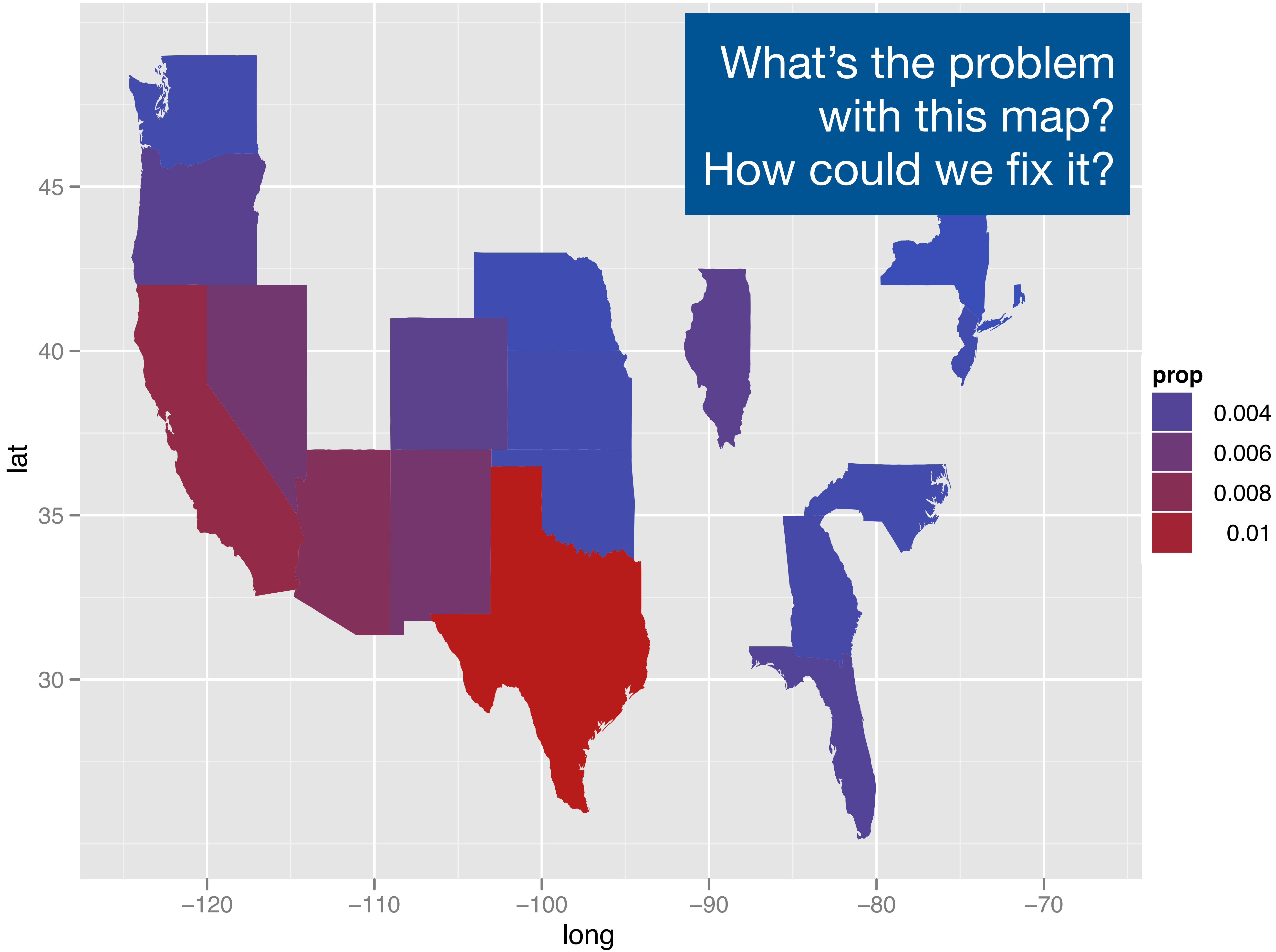
Spatial plots

Choropleth map:
map colour of areas to value.

Proportional symbol map:
map size of symbols to value

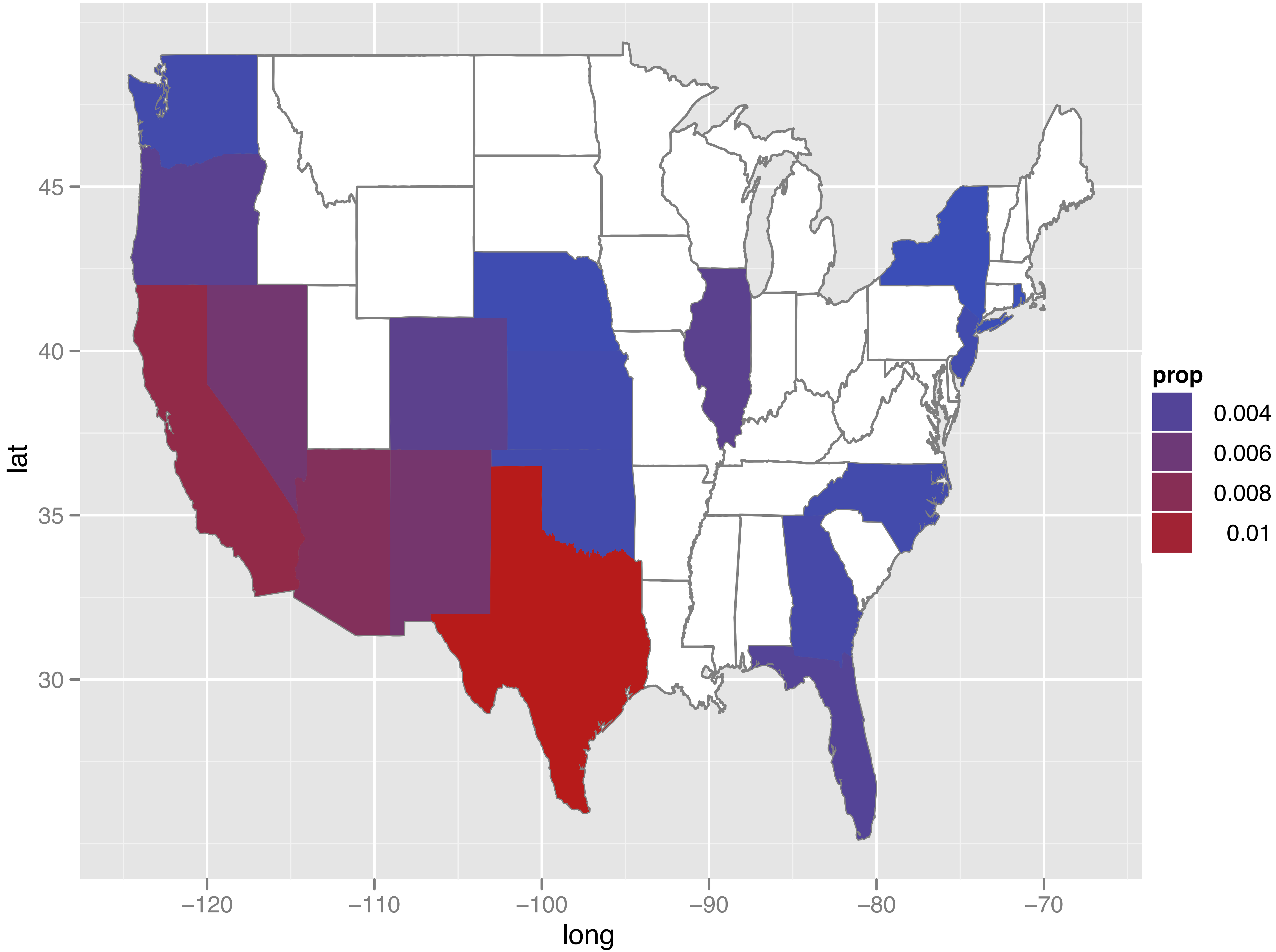
```
juan2000 <- subset(bnames, name == ”Juan” & year == 2000)

# Turn map data into normal data frame
library(maps)
states <- map_data(”state”)
states$state <- state.abb[match(states$region,
  tolower(state.name))]

# Merge and then restore original order
choropleth <- merge(juan2000, states,
  by = ”state”, all.y = T)
choropleth <- choropleth[order(choropleth$order), ]

# Plot with polygons
qplot(long, lat, data = choropleth, geom = ”polygon”,
  fill = prop, group = group)
```


45
40
35
30
lat
−120
−110
−100
−90
−80
−70
long
prop
0.004
0.006
0.008
0.01



What’s the problem with this map?
How could we fix it?

```
ggplot(choropleth, aes(long, lat, group = group)) +
  geom_polygon(fill = ”white”, colour = ”grey50”) +
  geom_polygon(aes(fill = prop))
```


45
40
35
30
lat
–120
–110
–100
–90
–80
–70
long
prop
0.004
0.006
0.008
0.01

Problems?

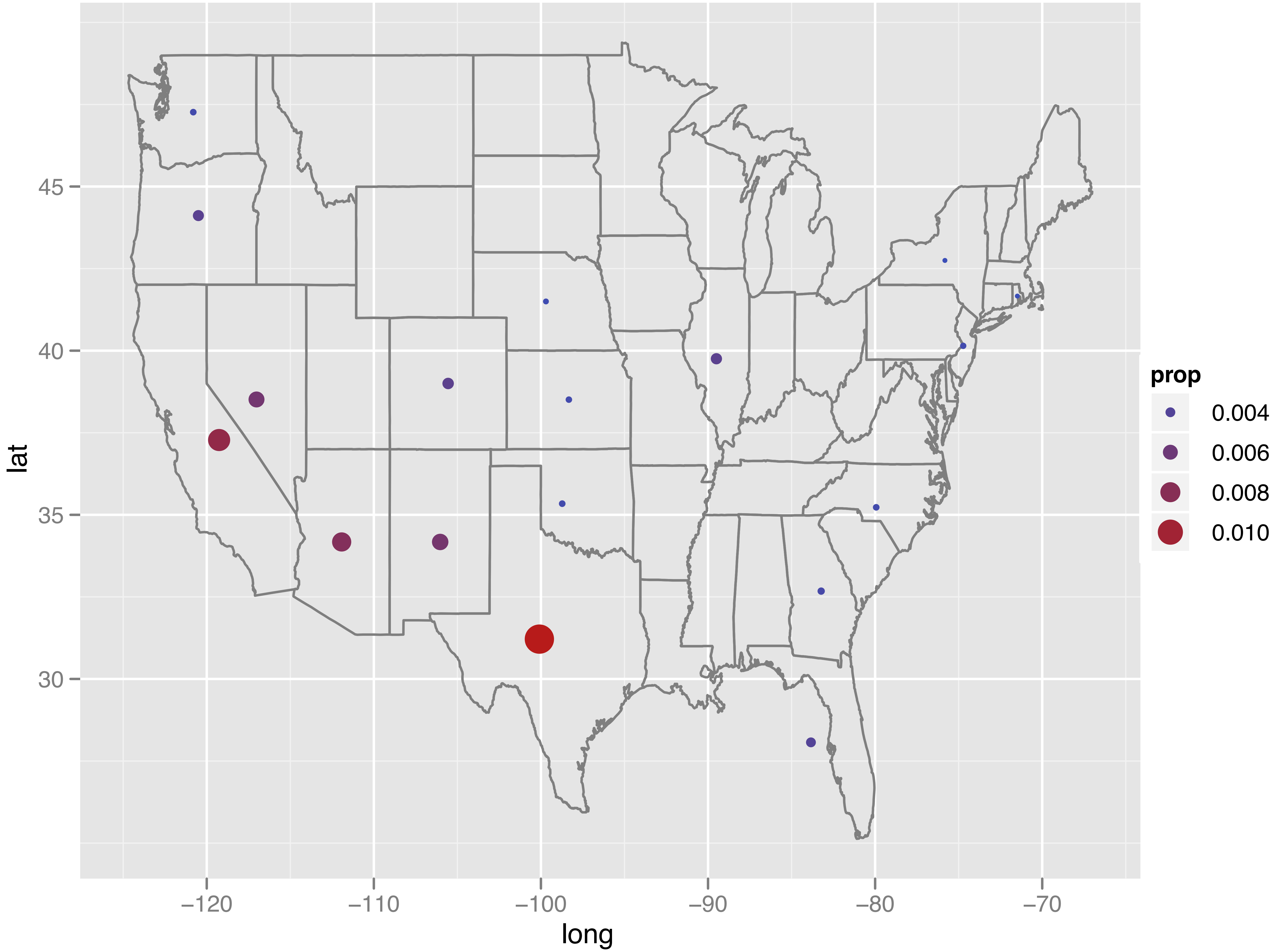
What are the problems with this sort of plot?

Take one minute to brainstorm some possible issues.

Problems

Big areas most striking. But in the US (as with most countries) big areas tend to least populated. Most populated areas tend to be small and dense - e.g. the East coast.

(Another computational problem: need to push around a lot of data to create these plots)

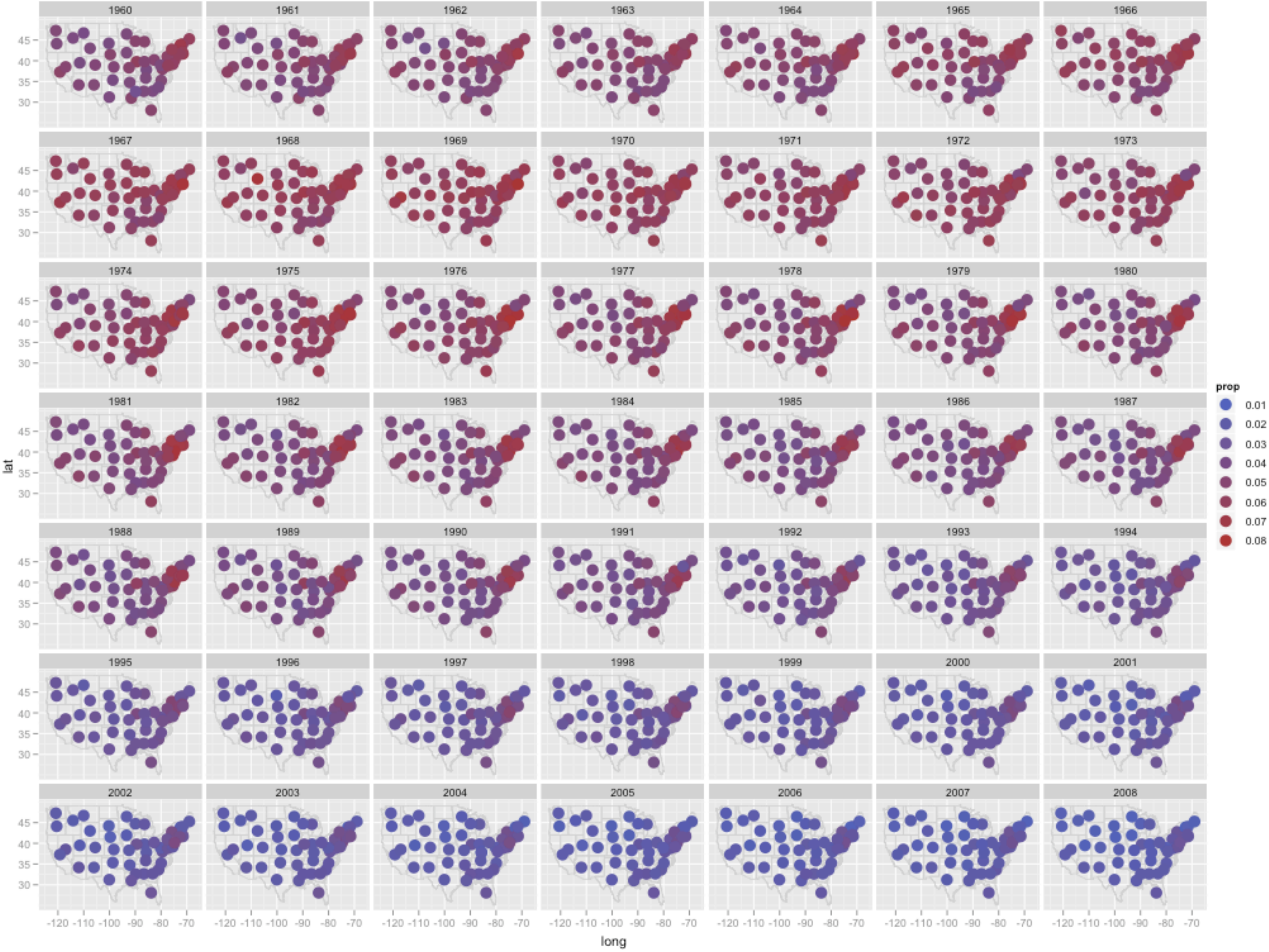

45
40
35
30
lat
−120
−110
−100
−90
−80
−70
long
prop
0.004
0.006
0.008
0.010

```
mid_range <- function(x) mean(range(x))
centres <- ddply(states, c(”state”), summarise,
  lat = mid_range(lat), long = mid_range(long))

bubble <- merge(juan2000, centres, by = ”state”)
qplot(long, lat, data = bubble,
  size = prop, colour = prop)

ggplot(bubble, aes(long, lat)) +
  geom_polygon(aes(group = group), data = states,
    fill = NA, colour = ”grey50”) +
  geom_point(aes(size = prop, colour = prop))
```

Your turn

Replicate either a choropleth or a proportional symbol map with the name of your choice.

Space | Time



Thursday, 22 October 2009

Your turn

Try and create this plot yourself. What is the main difference between this plot and the previous?

```
juan <- subset(bnames, name == ”Juan”)
bubble <- merge(juan, centres, by = ”state”)

ggplot(bubble, aes(long, lat)) +
  geom_polygon(aes(group = group), data = states,
    fill = NA, colour = ”grey50”) +
  geom_point(aes(size = prop, colour = prop)) +
  facet_wrap(~ year)
```

Aside: geographic data

Boundaries for most countries available from: http://gadm.org

To use with ggplot2, use the fortify function to convert to usual data frame.

Will also need to install the sp package.

```
# install.packages(”sp”)

library(sp)
load(url(”http://gadm.org/data/rda/CHE_adm1.RData”))

head(as.data.frame(gadm))
ch <- fortify(gadm, region = ”ID_1”)
str(ch)

qplot(long, lat, group = group, data = ch,
  geom = ”polygon”, colour = I(”white”))
```