Visualising time and space

Hadley Wickham

Assistant Professor / Dobelman Family Junior Chair

Department of Statistics / Rice University

June 2012

1. New data: baby names by state
2. Visualise time conditional on space
3. Merging data
4. Visualise space
5. Visualise space conditional on time

Baby names by state

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**.

Subset

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

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

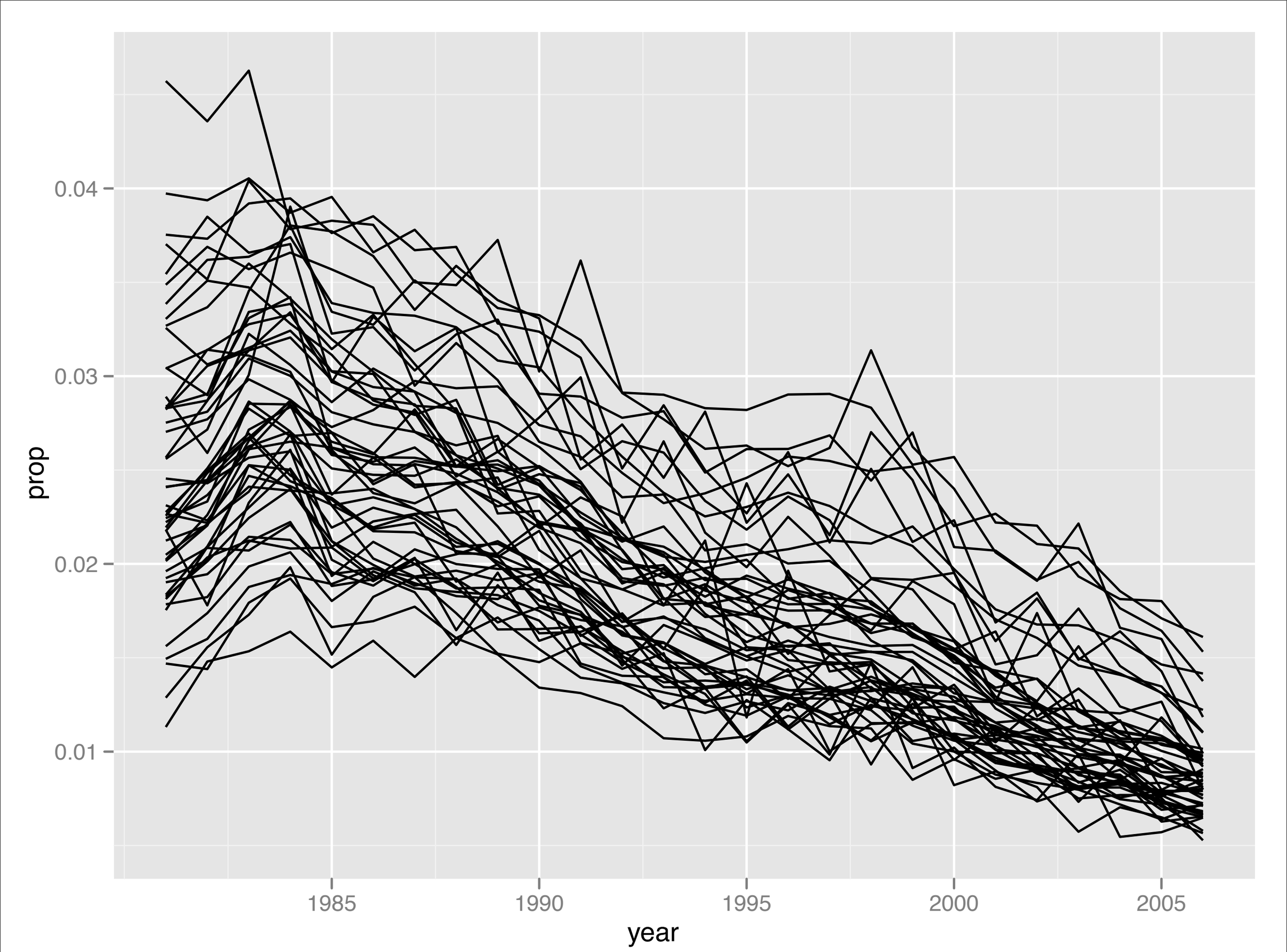
Aaron Alex Allison Alyssa Angela Ashley Carlos Chelsea Christian Eric Evan Gabriel Jacob Jared Jennifer Jonathan Juan Katherine Kelsey Kevin Matthew Michelle Natalie Nicholas Noah Rebecca Sara Sarah Taylor Thomas

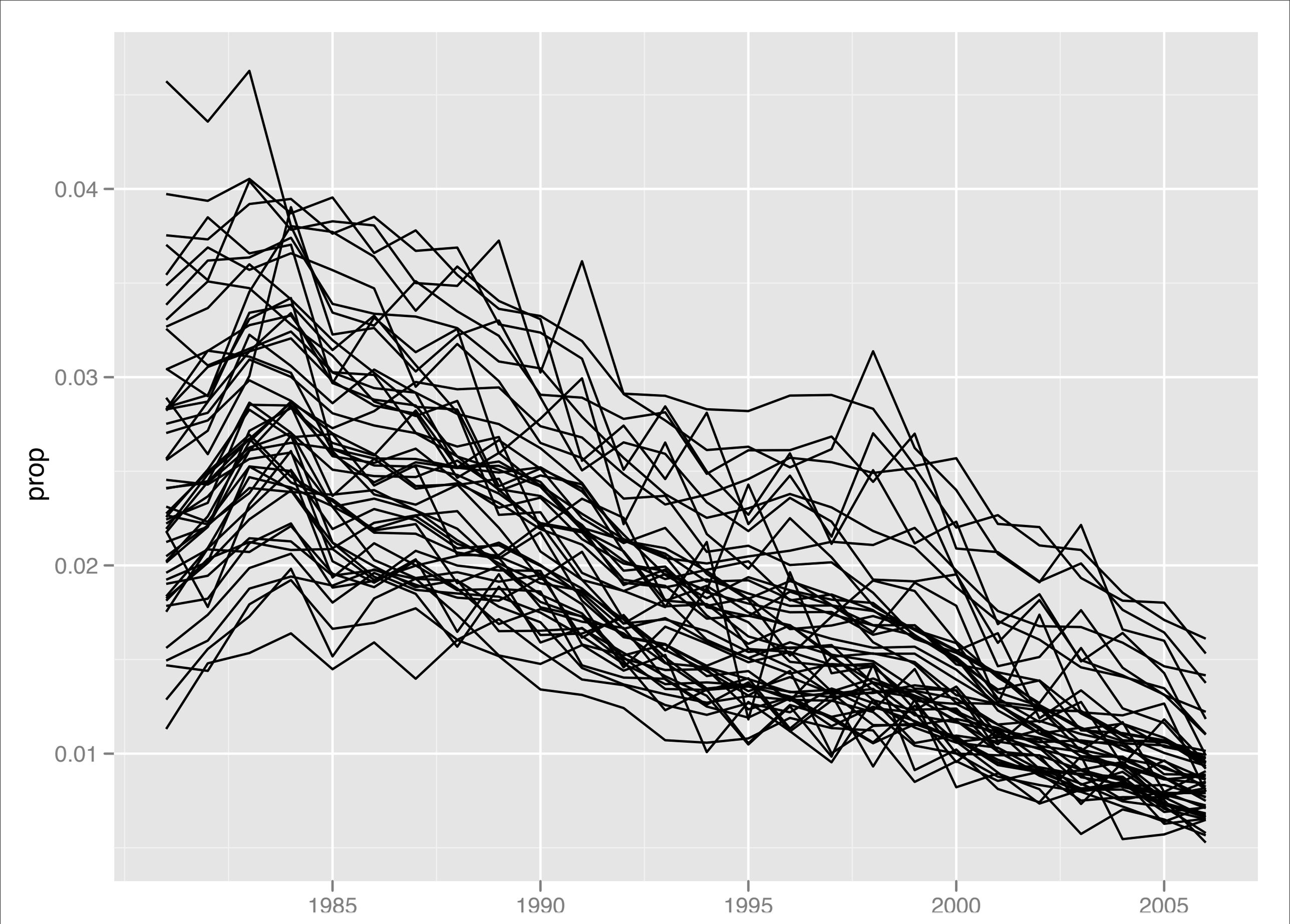
Getting started

```
library(ggplot2)
library(plyr)
```

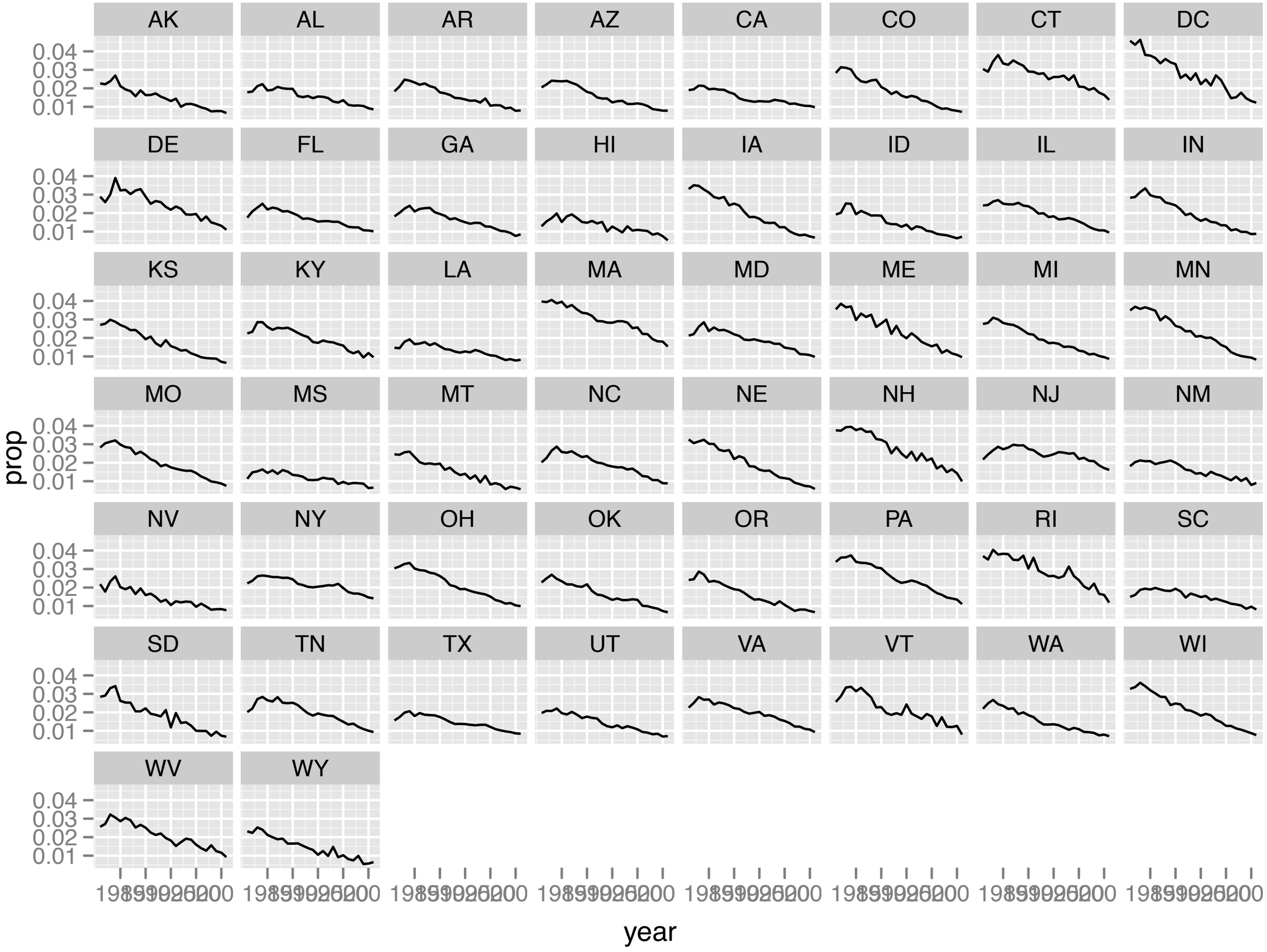
```
bnames <- read.csv("interesting-names.csv",
  stringsAsFactors = F)

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

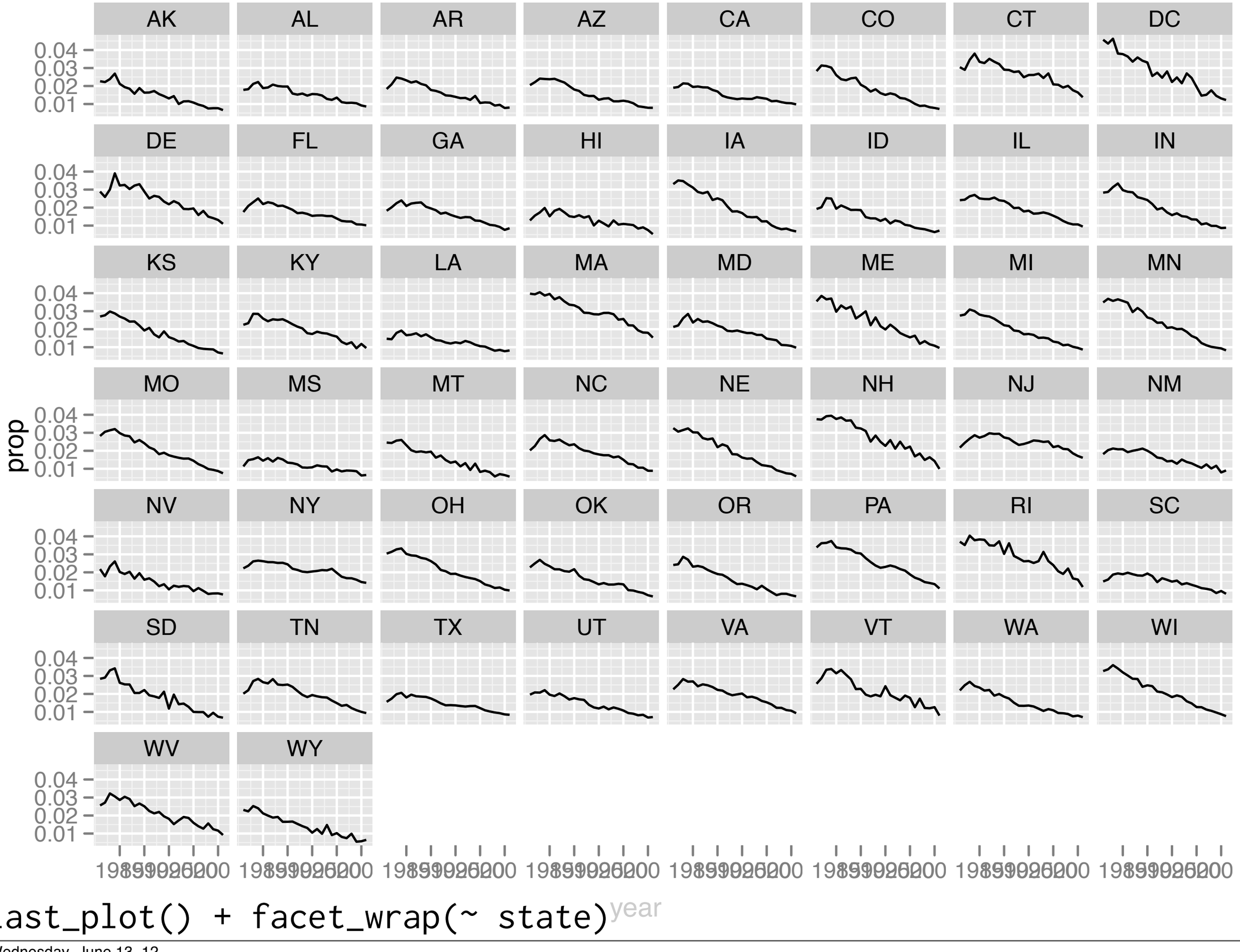
Time | Space


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
1980
1990
2000
year

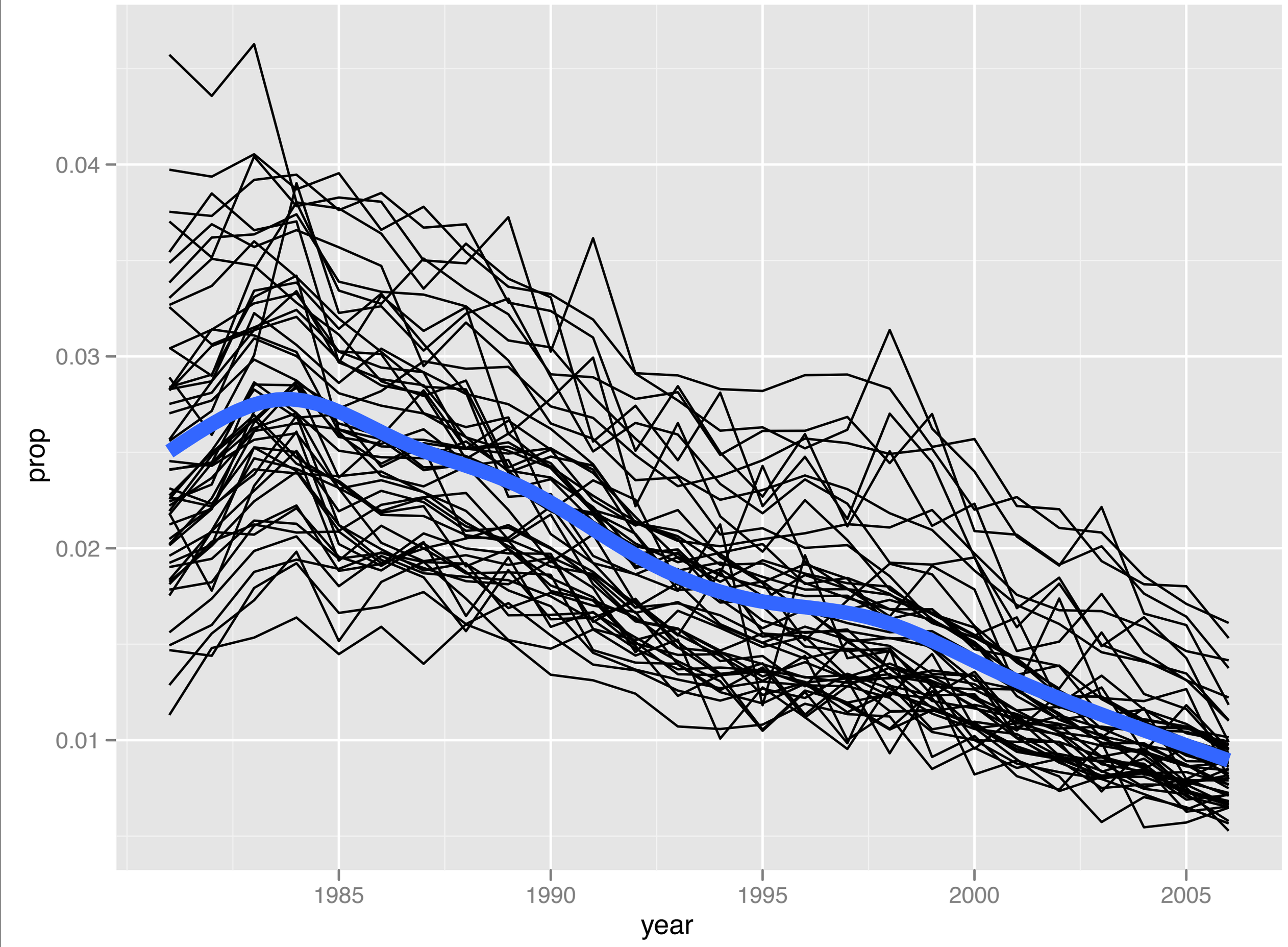

AK
AL
AR
AZ
CA
CO
CT
DC
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY
0.04
0.03
0.02
0.01
prop
1985
1990
1995
2000
year
last_plot() + facet_wrap(~ state)

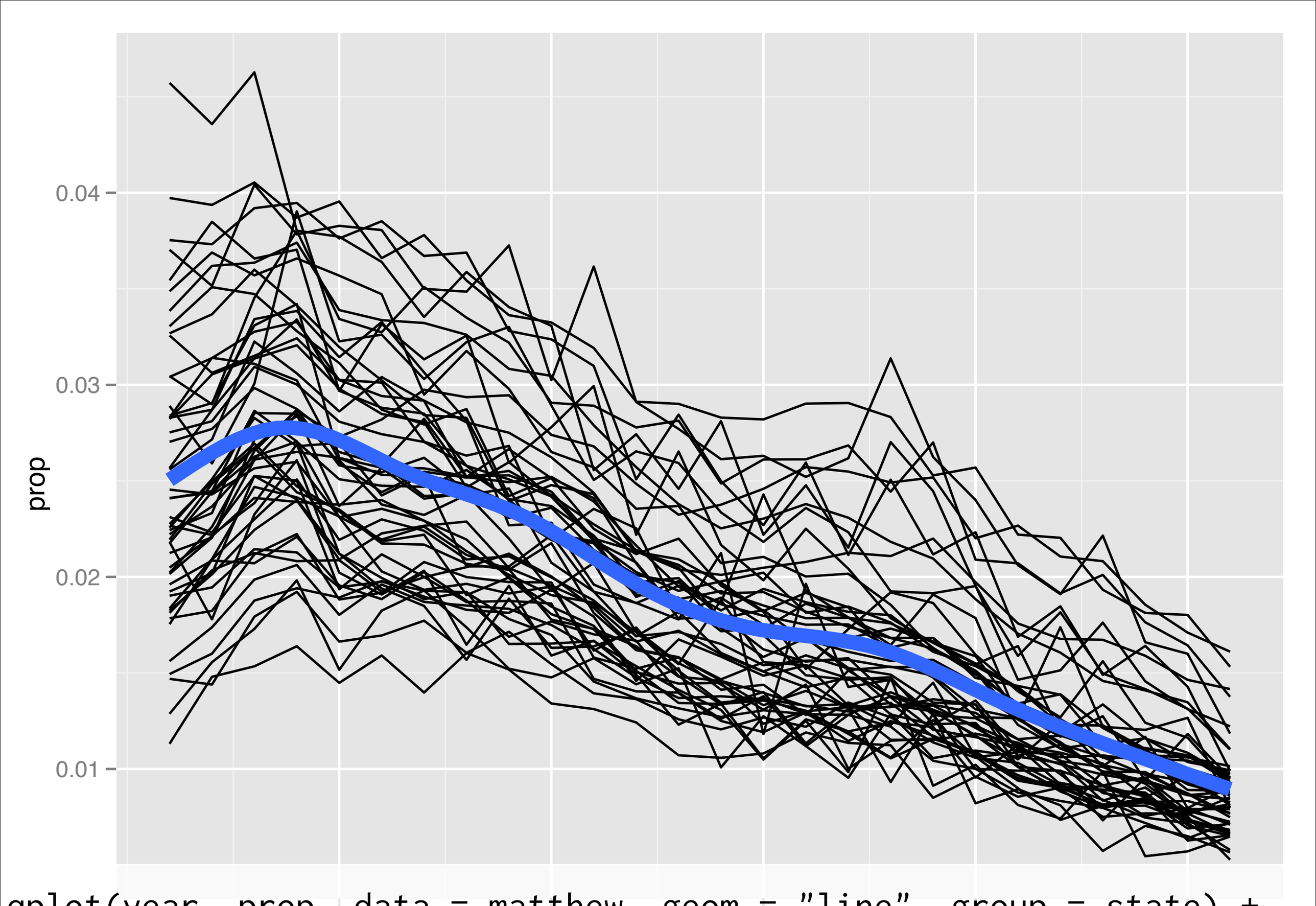
Your turn

Ensure that you can re-create these plots for other names. What do you see?

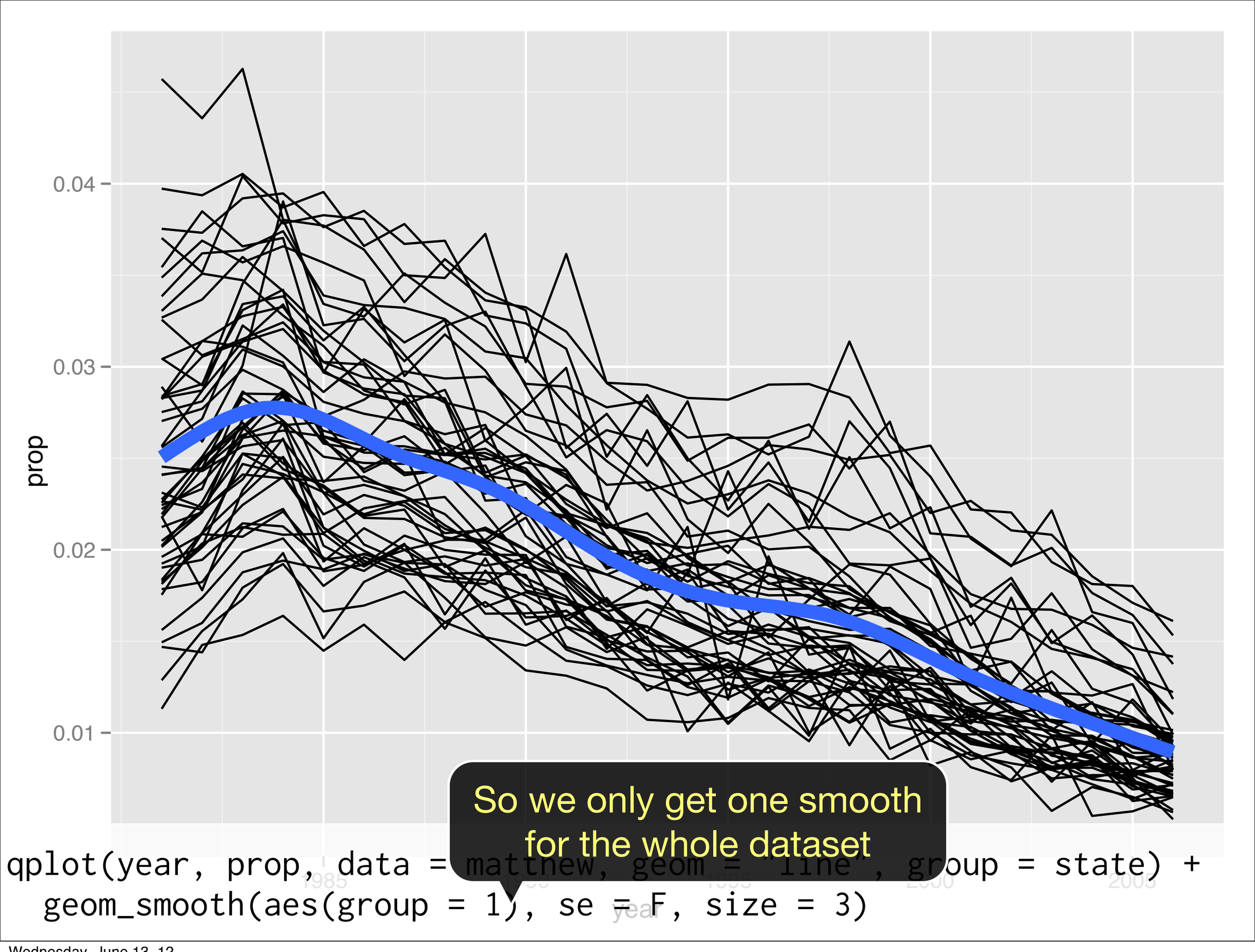
What names have unusual patterns?

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

show_name("Jennifer")
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)
```




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

Smoothing, Centering and Scaling

Three useful tools

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

Centering: remove differences in center by subtracting mean

Scaling: remove differences in range by dividing by sd, or by scaling to [0, 1]

```
library(mgcv)
smooth <- function(y, x, amount = 0.1) {
  mod <- gam(y ~ s(x, bs = "cr"), sp = amount)
  as.numeric(predict(mod))
}

matthew <- ddply(matthew, "state", mutate,
  prop_s1 = smooth(prop, year, amount = 0.01),
  prop_s2 = smooth(prop, year, amount = 0.1),
  prop_s3 = smooth(prop, year, amount = 1),
  prop_s4 = smooth(prop, year, amount = 10))

qplot(year, prop_s1, data = matthew, geom = "line",
  group = state)
```

```
ggplot(matthew, aes(year, group = state)) +
  geom_line(aes(y = prop_s1, colour = "s1")) +
  geom_line(aes(y = prop_s2, colour = "s2")) +
  geom_line(aes(y = prop_s3, colour = "s3")) +
  geom_line(aes(y = prop_s4, colour = "s4")) +
  facet_wrap(~ state)
```

```
center <- function(x) x - mean(x, na.rm = T)

matthew <- ddply(matthew, "state", mutate,
  prop_c = center(prop),
  prop_sc = center(prop_s1))

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

```
scale <- function(x) x / sd(x, na.rm = T)
scale01 <- function(x) {
  rng <- range(x, na.rm = T)
  (x - rng[1]) / (rng[2] - rng[1])
}
```

```
matthew <- ddply(matthew, "state", mutate,
  prop_ss = scale01(prop_s1))

qplot(year, prop_ss, 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"),
  mutate, 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(~ name, scales = "free_y")
```

Merging data

Combining datasets

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

?

```
what_played <- data.frame(name = c("John", "Paul",
  "George", "Ringo", "Stuart", "Pete"), instrument =
  c("guitar", "bass", "guitar", "drums", "bass",
  "drums"))
```

```
members <- data.frame(name = c("John", "Paul",
  "George", "Ringo", "Brian"), band = c("TRUE",
  "TRUE", "TRUE", "TRUE", "FALSE"))
```

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Stuart	bass	NA
Pete	drums	NA

```
join(x, y, type = "left")
```

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Stuart	bass	NA
Pete	drums	NA

```
join(x, y, type = "left")
```

Try it

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Brian	NA	F

```
join(x, y, type = "right")
```

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Brian	NA	F

```
join(x, y, type = "right")
```

Try it

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T

```
join(x, y, type = "inner")
```

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T

```
join(x, y, type = "inner")
```

Try it

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Stuart	bass	NA
Pete	drums	NA
Brian	NA	F

```
join(x, y, type = "full")
```

x

Name	instrument
John	guitar
Paul	bass
George	guitar
Ringo	drums
Stuart	bass
Pete	drums

+

y

Name	band
John	T
Paul	T
George	T
Ringo	T
Brian	F

=

Name	instrument	band
John	guitar	T
Paul	bass	T
George	guitar	T
Ringo	drums	T
Stuart	bass	NA
Pete	drums	NA
Brian	NA	F

```
join(x, y, type = "full")
```

Try it

Type	Action
`"left"`	Include all of x, and matching rows of y
`"right"`	Include all of y, and matching rows of x
`"inner"`	Include only rows in **both** x and y
`"full"`	Include all rows

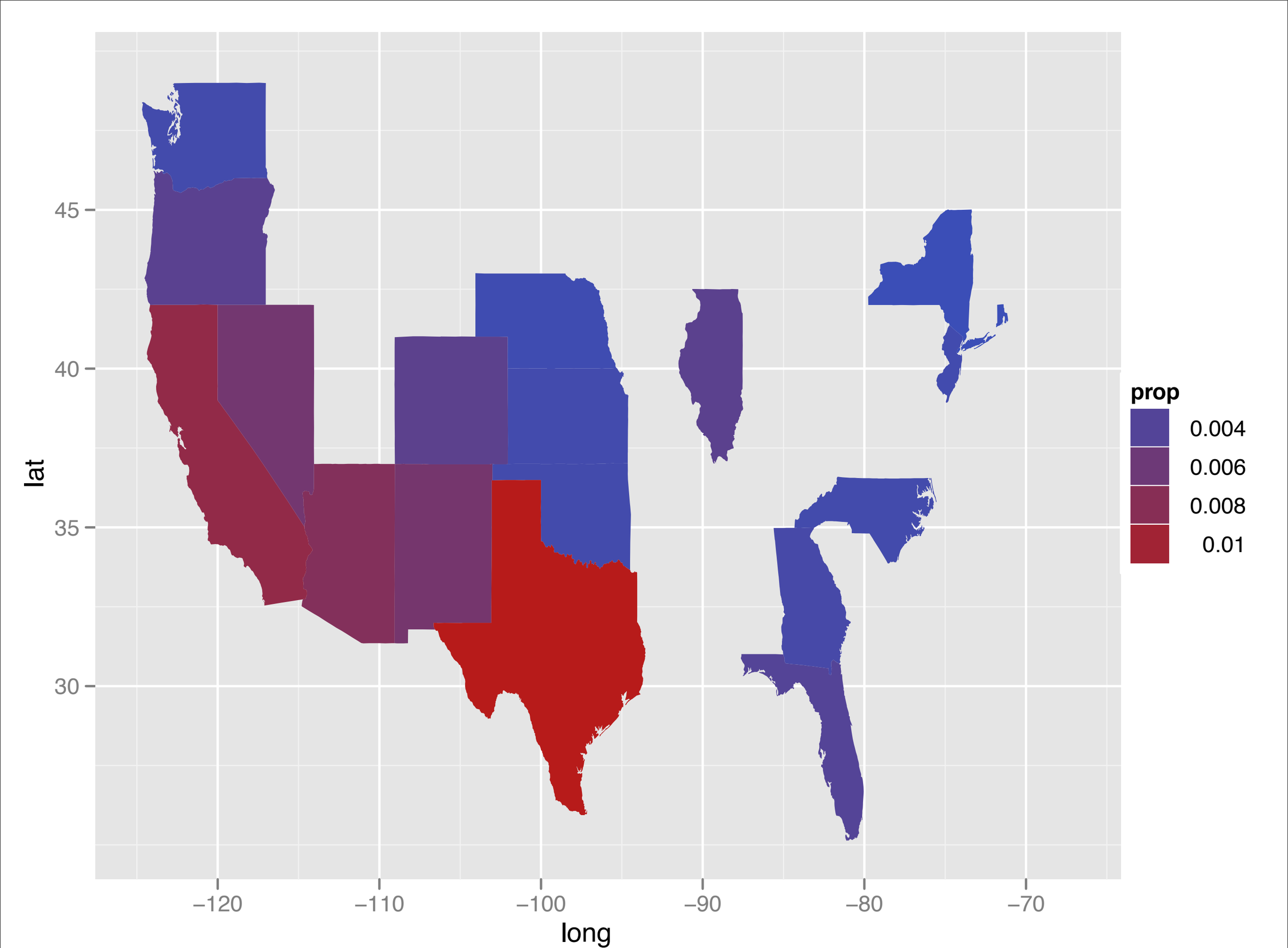
Space

Spatial plots

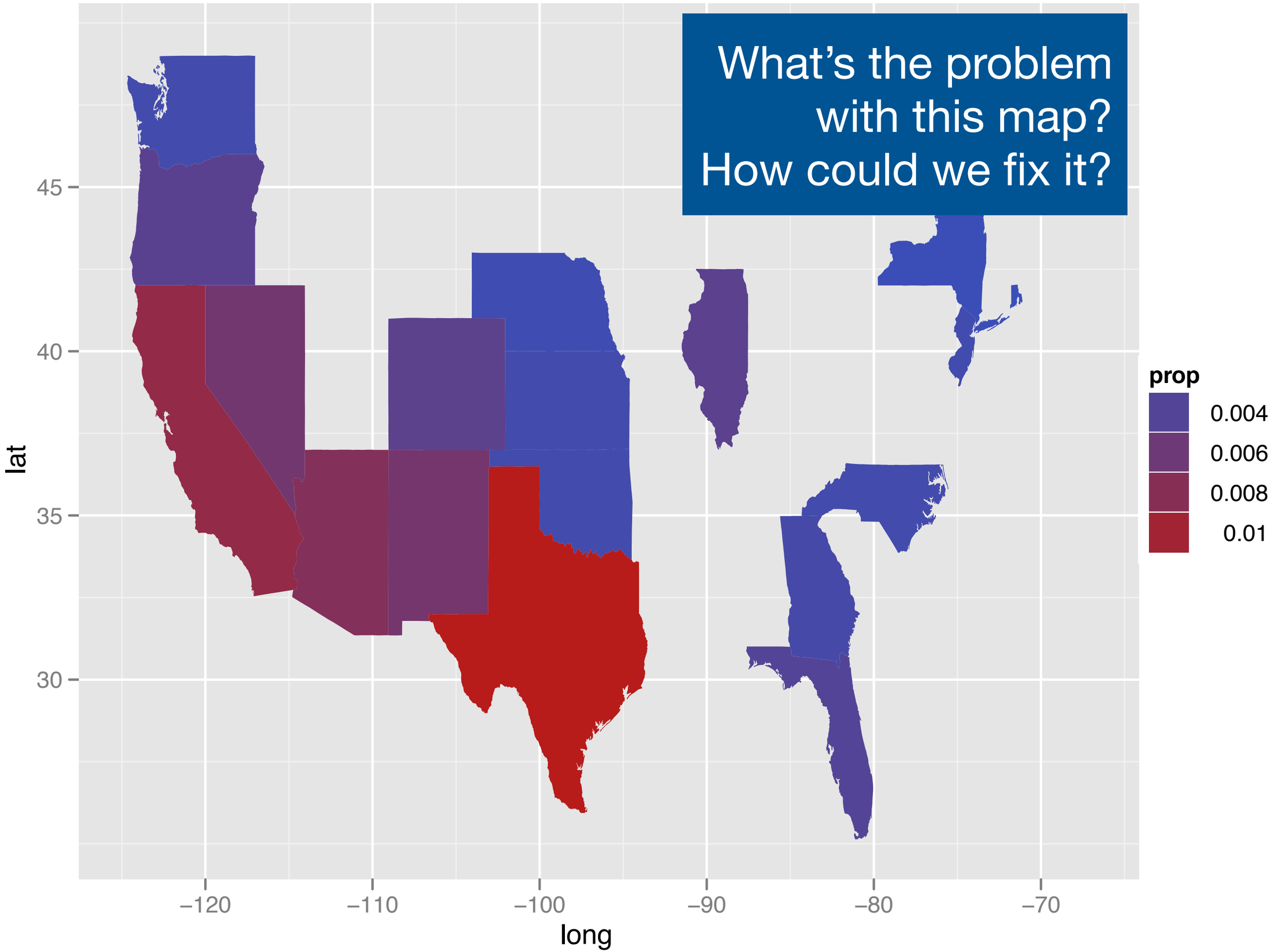
Choropleth map:
map colour of areas to value.

Proportional symbol map:
map size of symbols to value

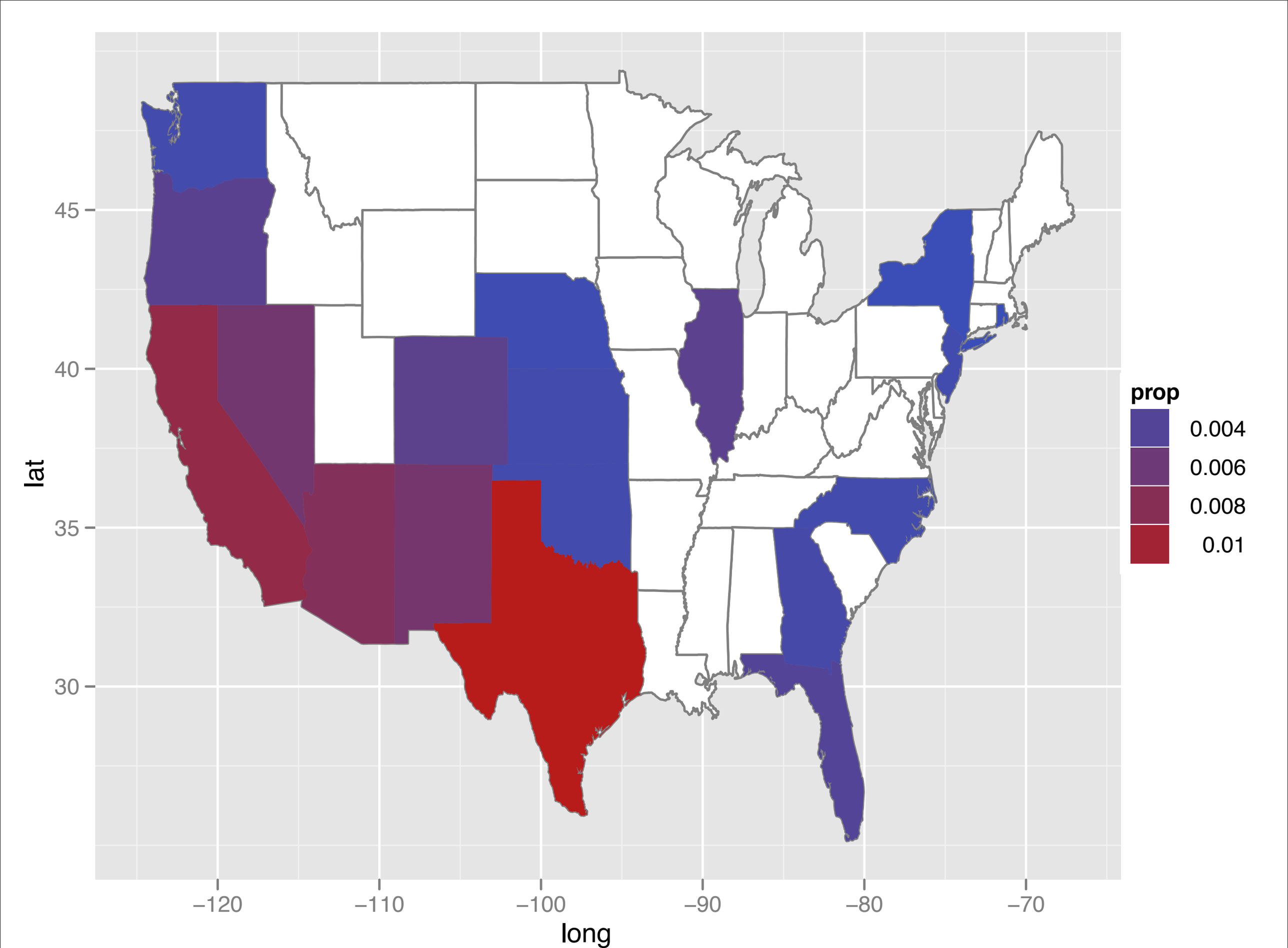
```
juan2000 <- subset(bnames, name == "Juan" &
  year == 2000)

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

# Join datasets
choropleth <- join(states, juan2000, by = "state")

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


lat
45
40
35
30
−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?

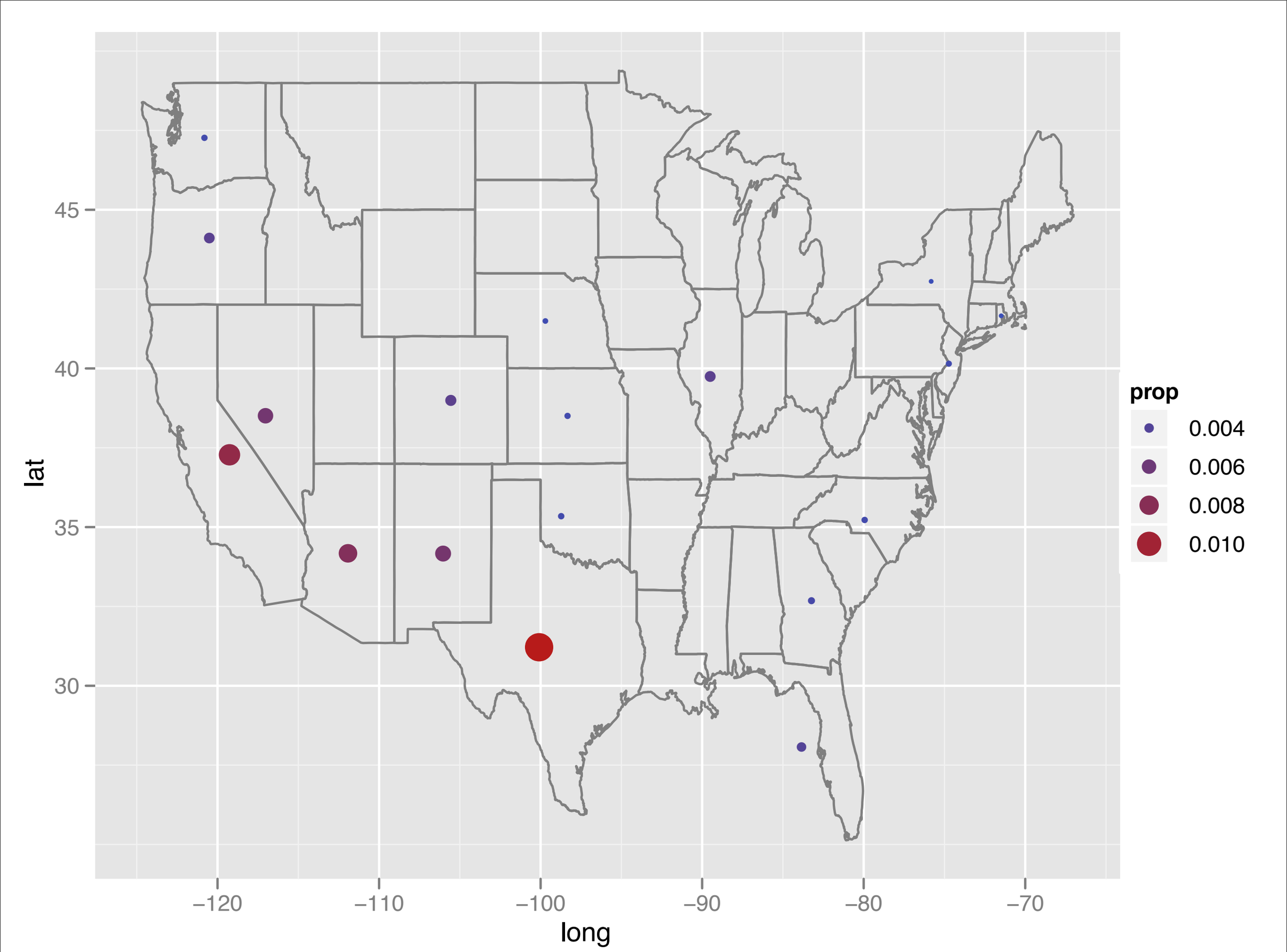
What are the problems with this sort of plot?

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)



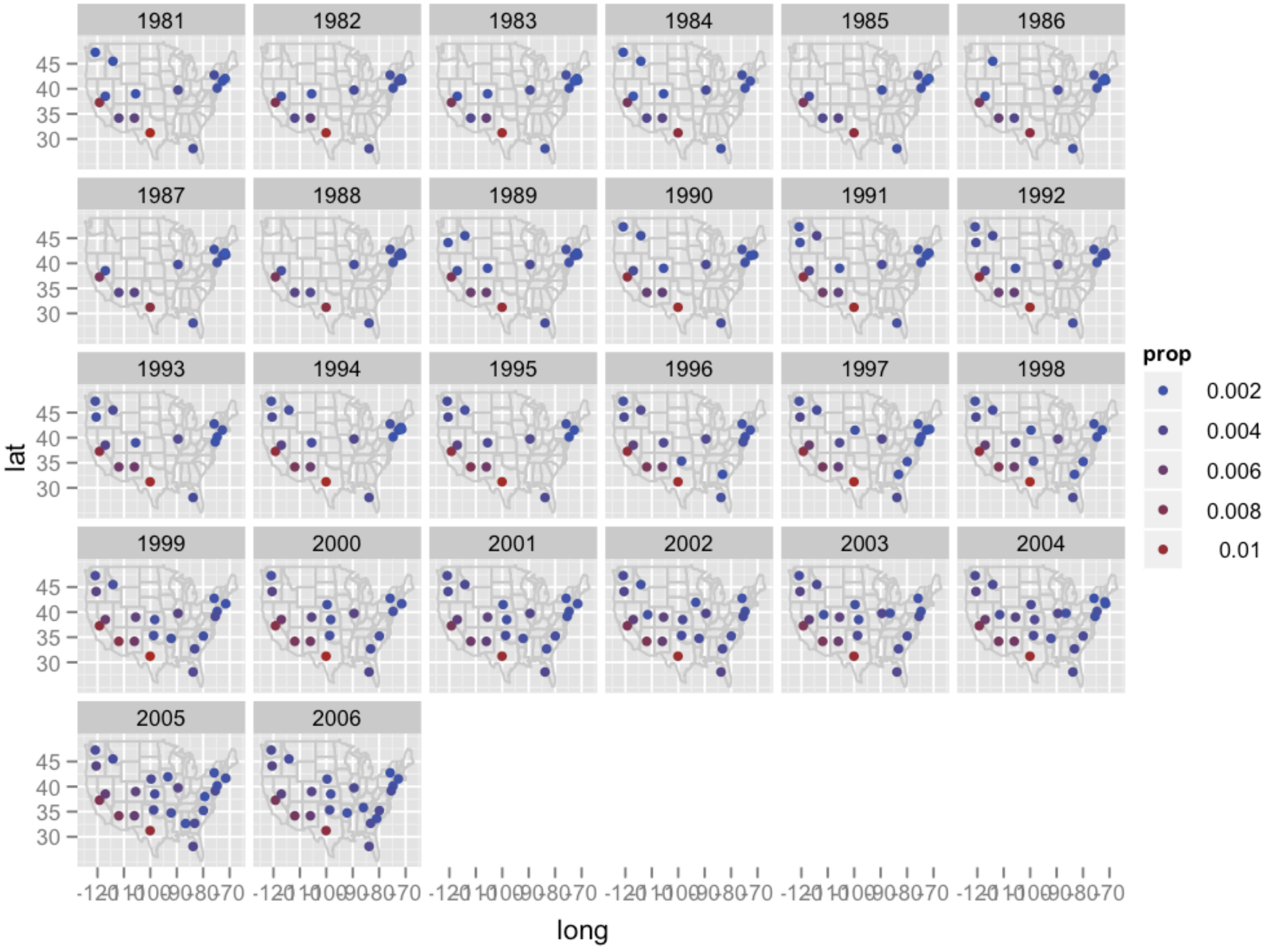
Wednesday, June 13, 12

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

bubble <- join(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


1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
45
40
35
30
lat
-120
-110
-100
-90
-80
-70
long
prop
0.002
0.004
0.006
0.008
0.01

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 <- join(juan, centres, by = "state")

ggplot(bubble, aes(long, lat)) +
  geom_polygon(aes(group = group), data = states,
    fill = NA, colour = "grey80") +
  geom_point(aes(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")) +
  coord_map()
```