SOC-5811 Week 3: Linear regression

Nick Graetz

University of Minnesota, Department of Sociology

9/22/2025

Load data

- R is an **object-oriented** programming language.

Load data

- R is an **object-oriented** programming language.
- We assign objects with the **<-** operator.

Load data

- R is an **object-oriented** programming language.
- We assign objects with the **<-** operator.
- We apply functions to objects: **function(object)**.

Load data

- R is an **object-oriented** programming language.
- We assign objects with the **<-** operator.
- We apply functions to objects: **function(object)**.
- All objects have a **class**.

Load data

- R is an **object-oriented** programming language.
- We assign objects with the **<-** operator.
- We apply functions to objects: **function(object)**.
- All objects have a **class**.
- All functions take inputs of certain classes and return outputs of certain classes.

Load data

- R is an **object-oriented** programming language.
- We assign objects with the **<-** operator.
- We apply functions to objects: **function(object)**.
- All objects have a **class**.
- All functions take inputs of certain classes and return outputs of certain classes.

- **Scripts are run in computing environments.**

Set up filepaths

- Assign my filepath to an object called "dropbox"
- Test different basic R functions

```
dropbox <- 'C:/Users/ngraetz/Dropbox/'
class(dropbox)
```

```
## [1] "character"
```

```
length(dropbox)
```

```
## [1] 1
```

```
nchar(dropbox)
```

```
## [1] 25
```

Set up filepaths

```
sum(dropbox)
```

```
## Error in sum(dropbox): invalid 'type' (character) of
```

Set up filepaths

- I'm going to set up a few filepaths.

```
fig_dir <- paste0(dropbox,
                  'Minnesota/repos/soc5811/figures/')
data_dir <- paste0(dropbox,
                   'Minnesota/repos/soc5811/data/')
```

Load data

- We are going to look at population and housing data from the 2000/2010 Census.

```
census <- read_dta(paste0(data_dir,
                          'state_pophouse.dta'))
```

Load data

```
class(census)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
dim(census)
```

```
## [1] 51  9
```

```
names(census)
```

```
## [1] "state"     "statefp"   "a00aa2000" "a00aa2010" "a41aa2000" "a41aa2010"
## [7] "pctpop"    "pcthouse"  "onepct"
```

```
head(census)
```

```
## # A tibble: 6 x 9
##   state   statefp a00aa2000 a00aa2010 a41aa2000 a41aa2010 pctpop pcthouse onepct
##   <chr>   <chr>       <dbl>     <dbl>     <dbl>     <dbl>  <dbl>    <dbl>  <dbl>
## 1 Alabama 01        4447100   4779736   1963711   2171853   7.48     10.6      0
## 2 Alaska  02         626932    710231    260978    306967  13.3      17.6      1
## 3 Arizona 04        5130632   6392017   2189189   2844526  24.6      29.9      2
## 4 Arkans~ 05        2673400   2915918   1173043   1316299   9.07     12.2      3
## 5 Califo~ 06       33871648  37253956  12214549  13680081   9.99     12.0      4
## 6 Colora~ 08        4301261   5029196   1808037   2212898  16.9      22.4      5
```

Examine data

We can use different functions like select() and slice() to look at specific rows and columns:

```
census %>%
  select(state,a00aa2000) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 2
##   state      a00aa2000
##   <chr>          <dbl>
## 1 Alabama      4447100
## 2 Alaska        626932
## 3 Arizona      5130632
## 4 Arkansas     2673400
## 5 California  33871648
```

Examine data

We can use other packages like "data.table" with different functions:

```
census <- as.data.table(census)
class(census)
```

```
## [1] "data.table" "data.frame"
```

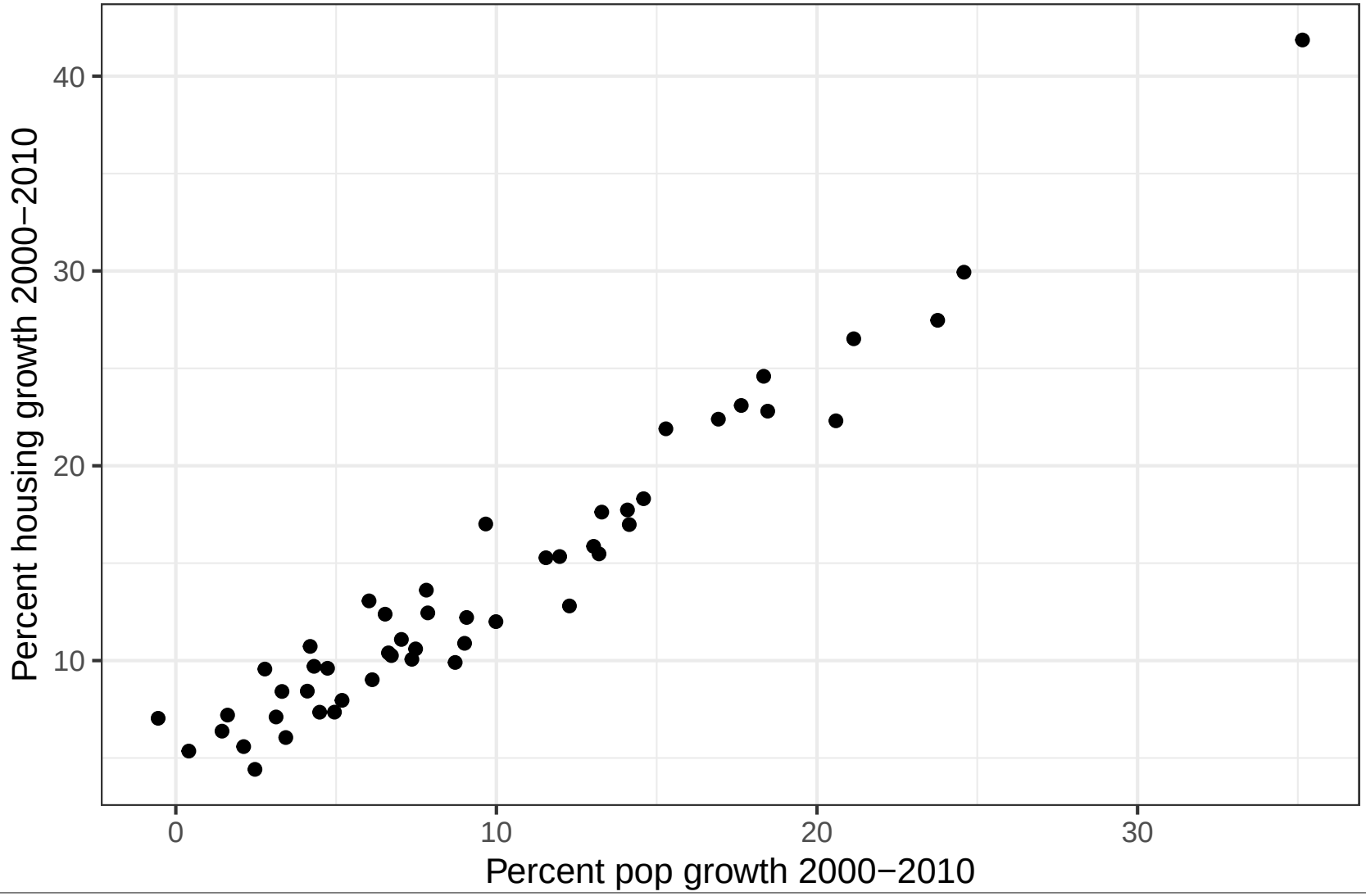
```
census[1:5, c('state','a00aa2000')]
```

```
##          state a00aa2000
##         <char>     <num>
## 1:    Alabama   4447100
## 2:     Alaska    626932
## 3:    Arizona   5130632
## 4:   Arkansas   2673400
## 5: California  33871648
```

Examine data

```
plot <- ggplot(data=census,
                aes(x=pctpop,
                    y=pcthouse)) +
  geom_point() +
  labs(x='Percent pop growth 2000-2010',
       y='Percent housing growth 2000-2010') +
  theme_bw()
```

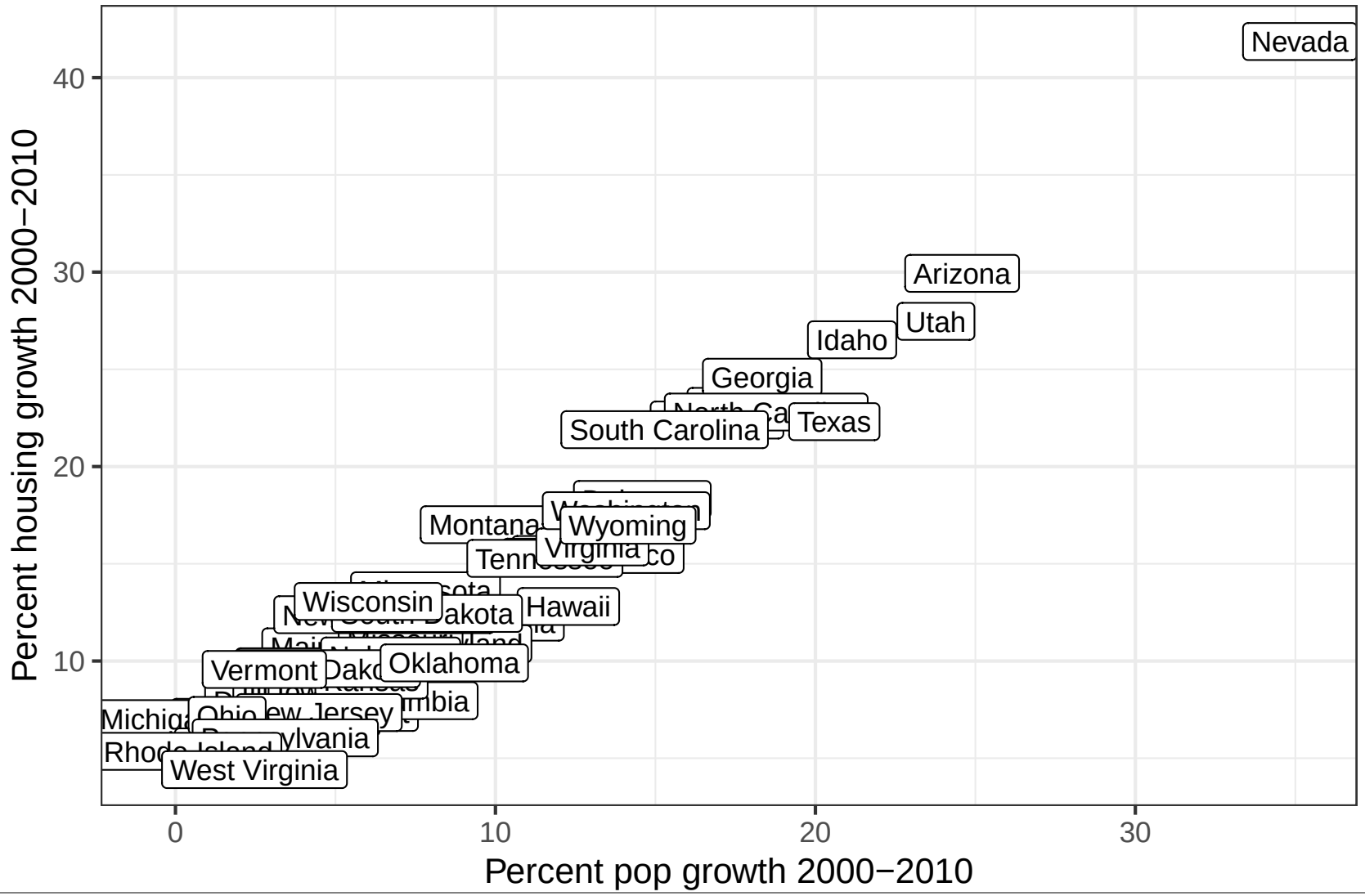
Examine data



Examine data

```
plot <- ggplot(data=census,
               aes(x=pctpop,
                   y=pcthouse,
                   label=state)) +
  geom_label(size=3) +
  labs(x='Percent pop growth 2000-2010',
       y='Percent housing growth 2000-2010') +
  theme_bw()
```

Examine data



Population regression functions

Let's think about creating a **model** for housing growth:

$$pcthouse = f(pctpop)$$

- What is a model?

Population regression functions

Let's think about creating a **model** for housing growth:

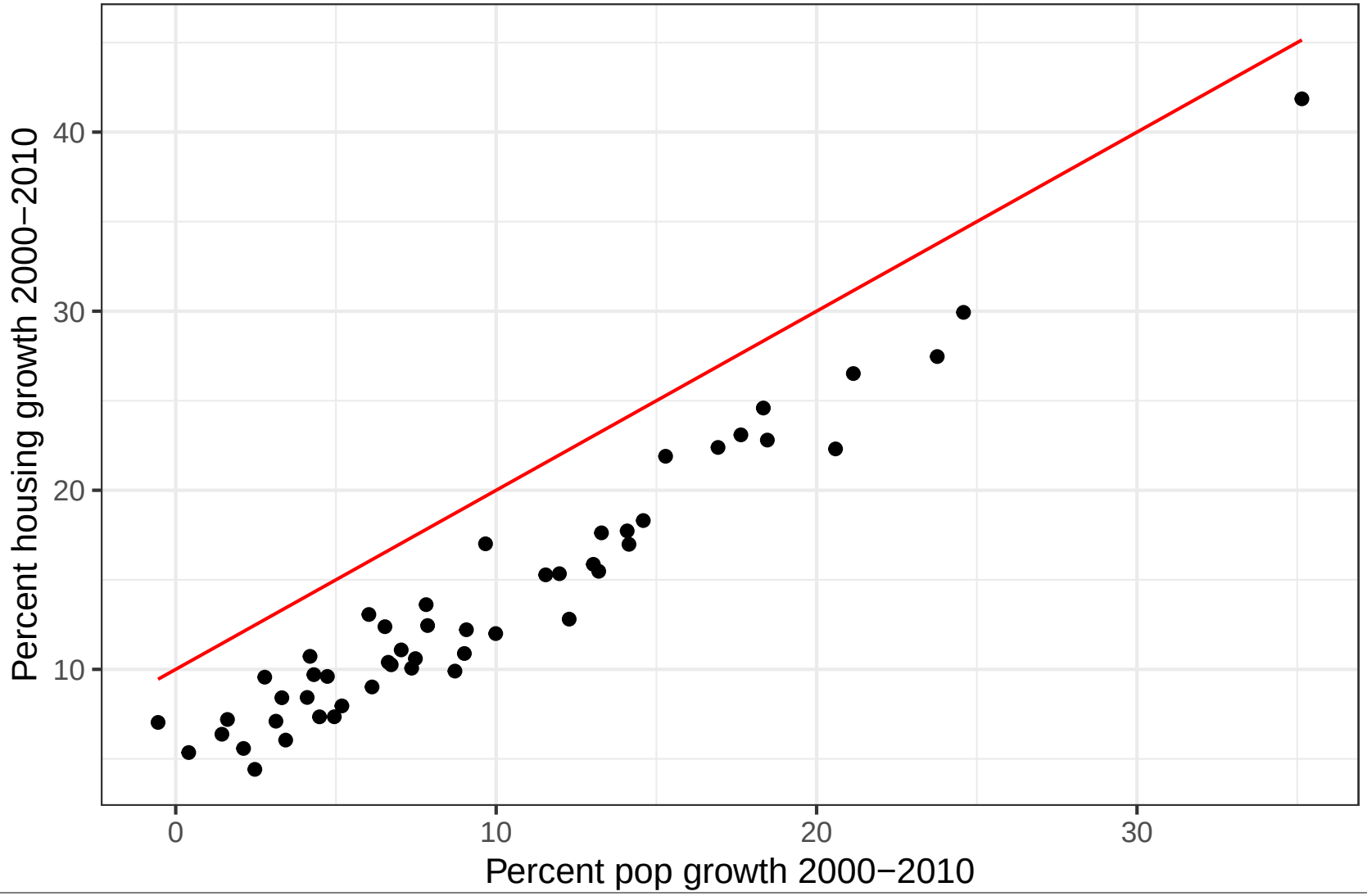
$$pcthouse = 10 + pctpop$$

- Models are defined by *coefficients* (or more generally, *parameters*).

Examine data

```
census <- census %>%
  mutate(pcthouse_mod1 = 10 + pctpop)
```

Examine data



Population regression functions

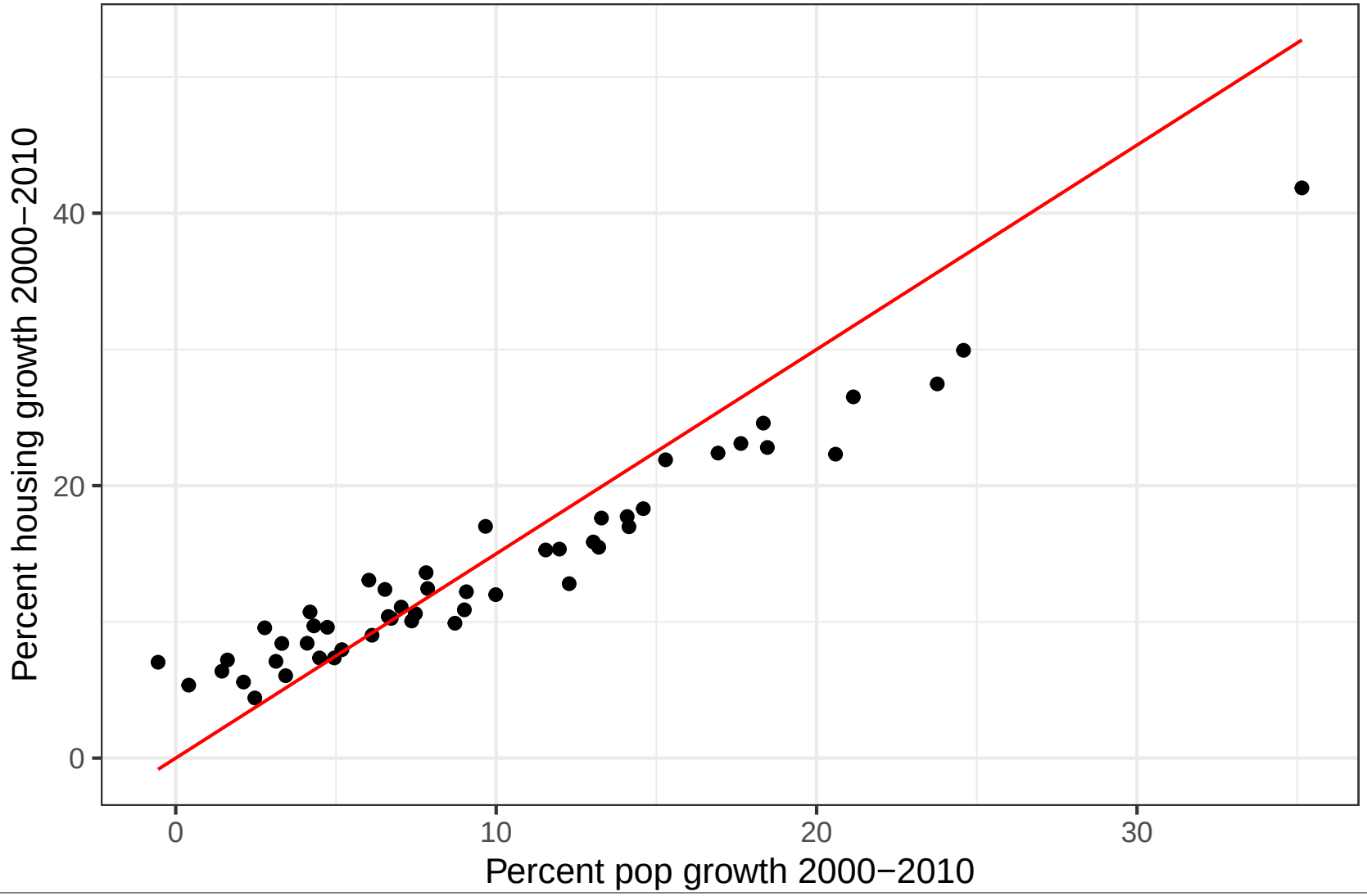
Let's think about creating a **model** for housing growth:

$$pcthouse = 1.5 \times pctpop$$

Examine data

```
census <- census %>%
  mutate(pcthouse_mod1 = 1.5 * pctpop)
```

Examine data



Population regression functions

- How do I pick a good model?
- What makes a model good?
- What is my goal?

$$pcthouse = f(pctpop)$$

Population regression functions

Let's think about creating a model for housing growth:

$$pcthouse = f(pctpop)$$

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

Population regression functions

Fitting a linear regression with data:

```
model <- lm(pcthouse~pctpop,
            data=census)
summary(model)
```

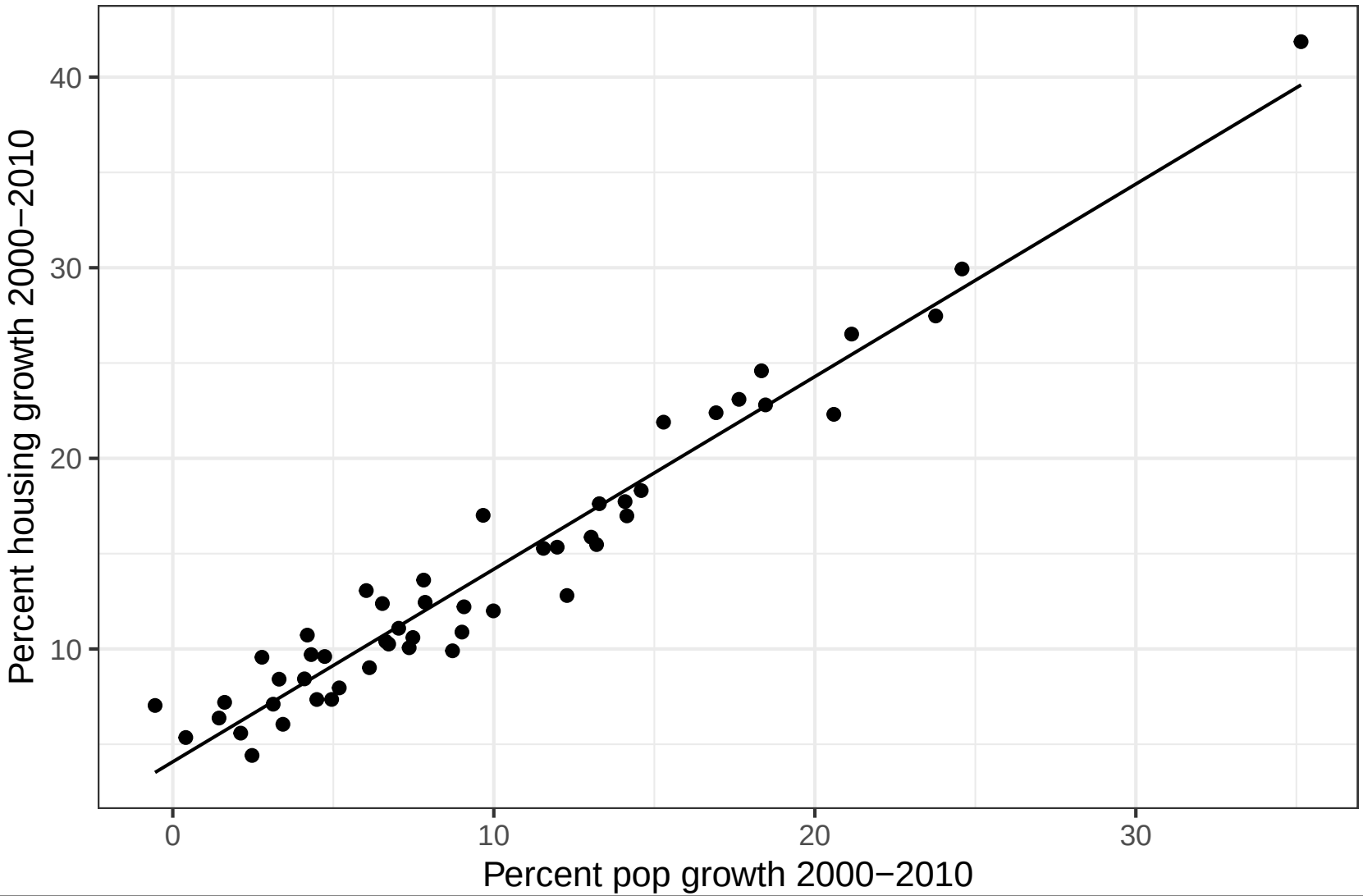
```
##
## Call:
## lm(formula = pcthouse ~ pctpop, data = census)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6830 -1.3132 -0.1364  1.2039  3.5126
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.08125    0.40793   10.01 1.98e-13 ***
## pctpop       1.01030    0.03371   29.97  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.719 on 49 degrees of freedom
## Multiple R-squared:  0.9483, Adjusted R-squared:  0.9472
## F-statistic: 898.1 on 1 and 49 DF,  p-value: < 2.2e-16
```

Making predictions

```
census <- census %>%
  mutate(pcthouse_pred=predict(model))
census %>%
  select(state,pctpop,pcthouse,pcthouse_pred) %>%
  head()
```

```
##          state    pctpop pcthouse pcthouse_pred
##         <char>     <num>    <num>         <num>
## 1:    Alabama  7.479841 10.59942      11.63810
## 2:     Alaska 13.286768 17.62179      17.50481
## 3:    Arizona 24.585373 29.93515      28.91974
## 4:   Arkansas  9.071520 12.21234      13.24616
## 5: California  9.985662 11.99825      14.16972
## 6:   Colorado 16.923758 22.39230      21.17924
```

Making predictions



Saving 6 x 4 in image

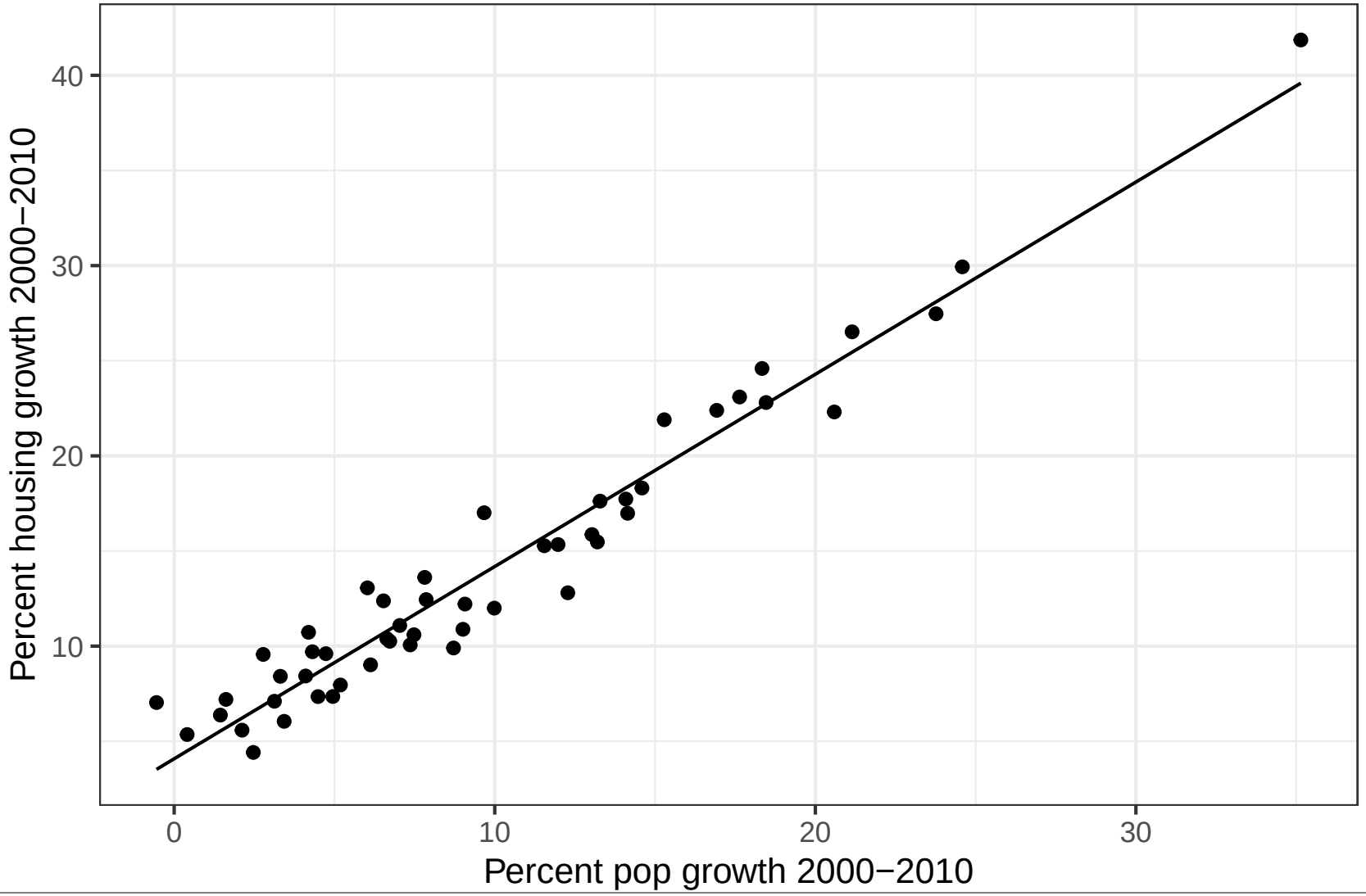
Model coefficients

- What does it mean to "fit" a regression model?
- How did R come up with the coefficients 4.08 and 1.01?

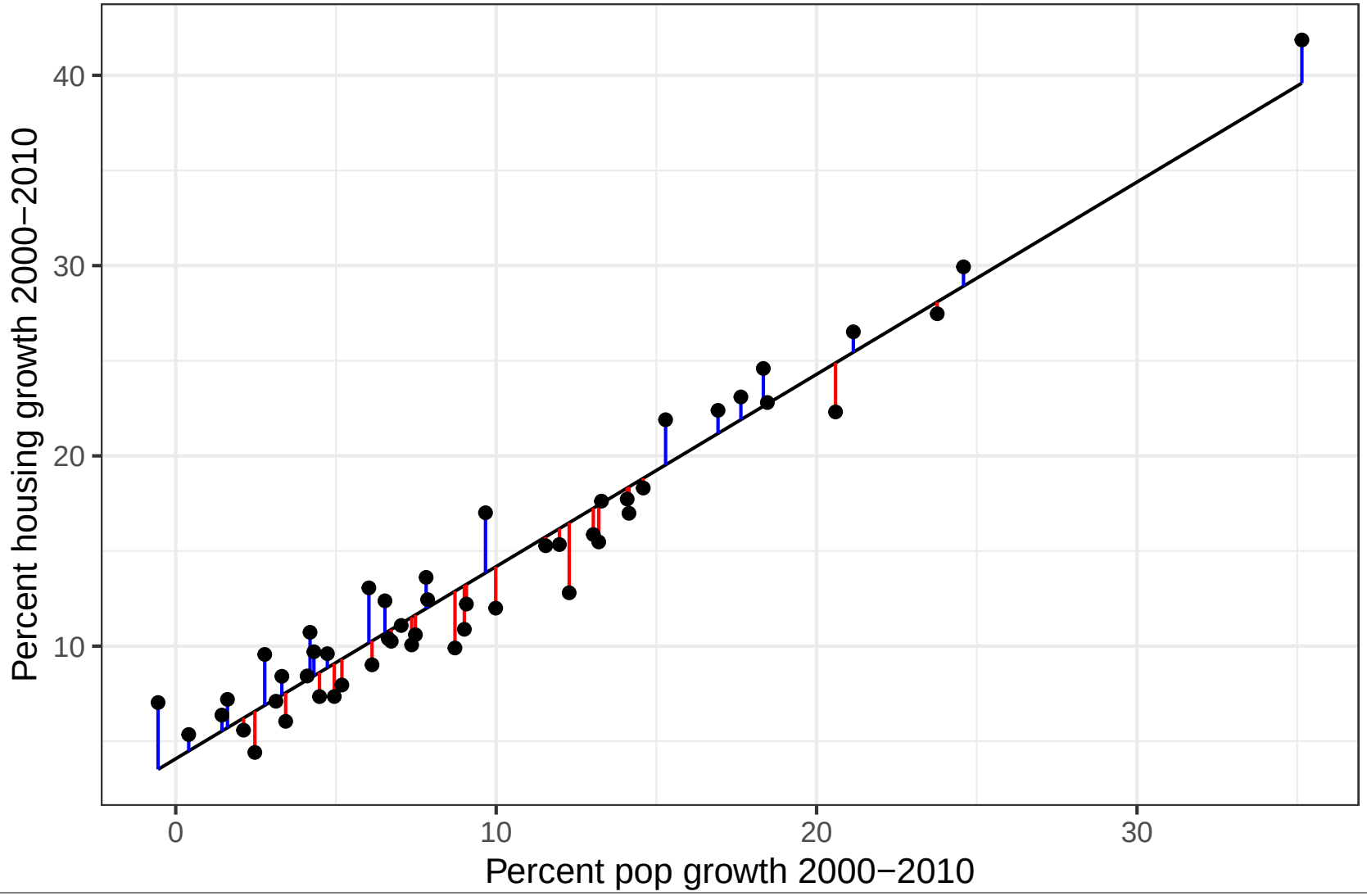
```
summary(model)
```

```
##
## Call:
## lm(formula = pcthouse ~ pctpop, data = census)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6830 -1.3132 -0.1364  1.2039  3.5126
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.08125    0.40793   10.01 1.98e-13 ***
## pctpop       1.01030    0.03371   29.97  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.719 on 49 degrees of freedom
## Multiple R-squared:  0.9483, Adjusted R-squared:  0.9472
## F-statistic: 898.1 on 1 and 49 DF,  p-value: < 2.2e-16
```

Making predictions



Making predictions



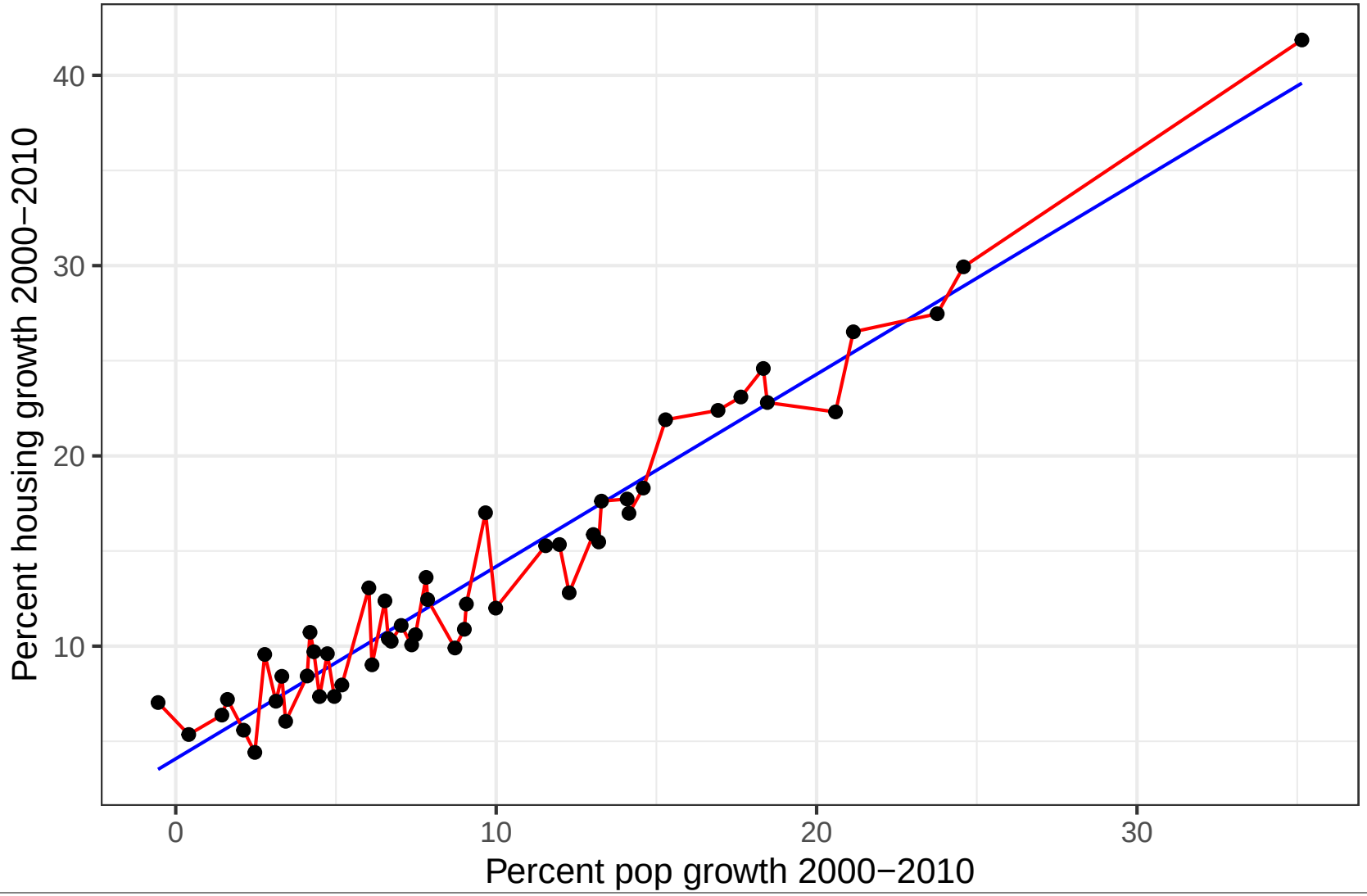
USING MODELS TO PREDICT

Just looking *within* my sample... why is my model always wrong?

```
census %>%
  select(state,pctpop,pcthouse,pcthouse_pred) %>%
  head()
```

```
##          state    pctpop pcthouse pcthouse_pred
##         <char>     <num>    <num>          <num>
## 1:     Alabama  7.479841 10.59942       11.63810
## 2:      Alaska 13.286768 17.62179       17.50481
## 3:     Arizona 24.585373 29.93515       28.91974
## 4:    Arkansas  9.071520 12.21234       13.24616
## 5: California  9.985662 11.99825       14.16972
## 6:    Colorado 16.923758 22.39230       21.17924
```

Using models to predict



Coefficients

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

$$pcthouse_i = 4.08 + 1.01 pctpop_i + \epsilon_i$$

- Coefficients represent **average comparisons**.

Coefficients

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

$$pcthouse_i = 4.08 + 1.01 pctpop_i + \epsilon_i$$

- Coefficients represent **average comparisons**.
- Interpreting the coefficient on *pctpop* (e.g., x):

Coefficients

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

$$pcthouse_i = 4.08 + 1.01 pctpop_i + \epsilon_i$$

- Coefficients represent **average comparisons**.
- Interpreting the coefficient on *pctpop* (e.g., x):
  - On average, a 1-point increase in x is associated with a 1.01-point increase in y.

Coefficients

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

$$pcthouse_i = 4.08 + 1.01 pctpop_i + \epsilon_i$$

- Coefficients represent **average comparisons**.
- Interpreting the coefficient on *pctpop* (e.g., x):
  - On average, a 1-point increase in x is associated with a 1.01-point increase in y.
  - Across all values of x, the average difference in y at x and x+1 is 1.01.

Coefficients

$$pcthouse_i = \beta_0 + \beta_1 pctpop_i + \epsilon_i$$

$$pcthouse_i = 4.08 + 1.01 pctpop_i + \epsilon_i$$

- Coefficients represent **average comparisons**.
- Interpreting the coefficient on *pctpop* (e.g., x):
  - On average, a 1-point increase in x is associated with a 1.01-point increase in y.
  - Across all values of x, the average difference in y at x and x+1 is 1.01.
  - The slope of the predicted line of y across all values of x is 1.01.

Using models to compare

- Regression is a mathematical tool for making predictions.

Using models to compare

- Regression is a mathematical tool for making predictions.
- Regression coefficients can *sometimes* be interpreted as effects.

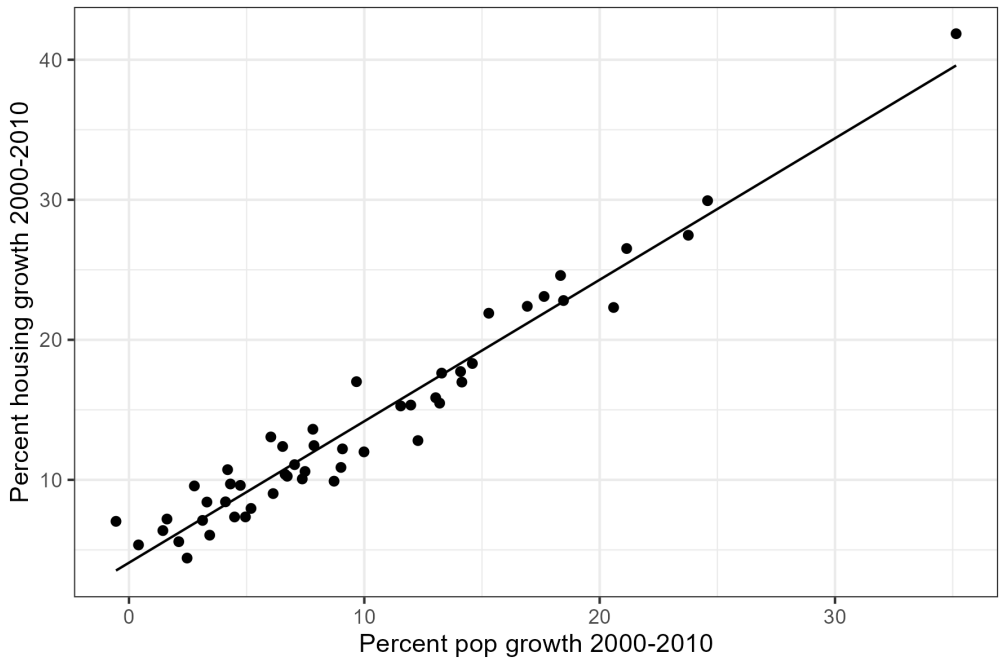
Using models to compare

- Regression is a mathematical tool for making predictions.
- Regression coefficients can *sometimes* be interpreted as effects.
- Regression coefficients can *always* be interpreted as average comparisons.

Building models

What can we do with this model?

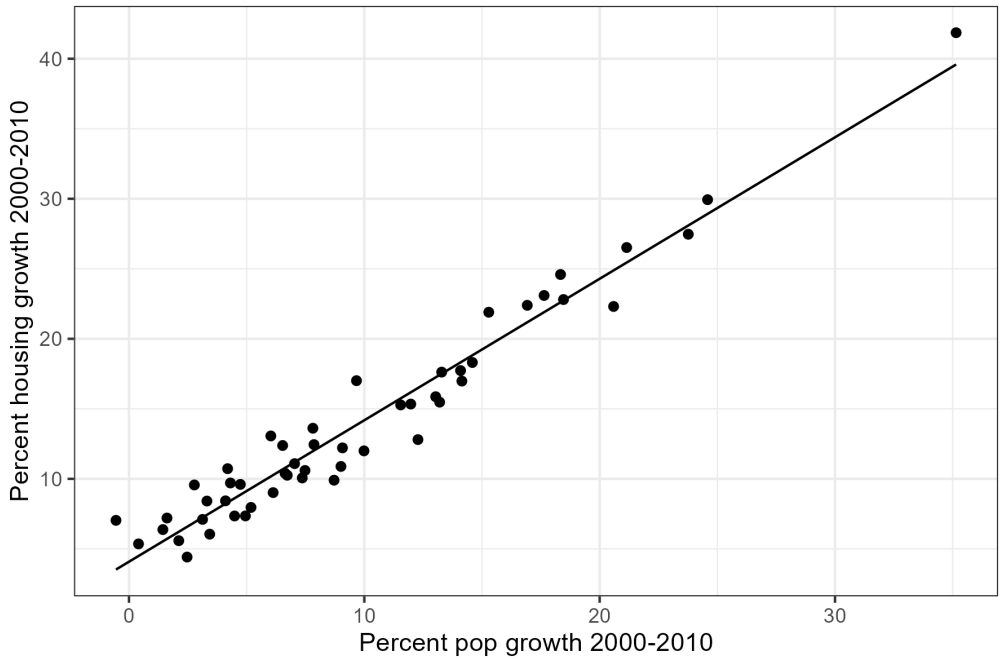
1. Generalizing from sample to population.
2. Measurement.
3. Forecasting.
4. Causal inference.



Building models

What can we do with this model?

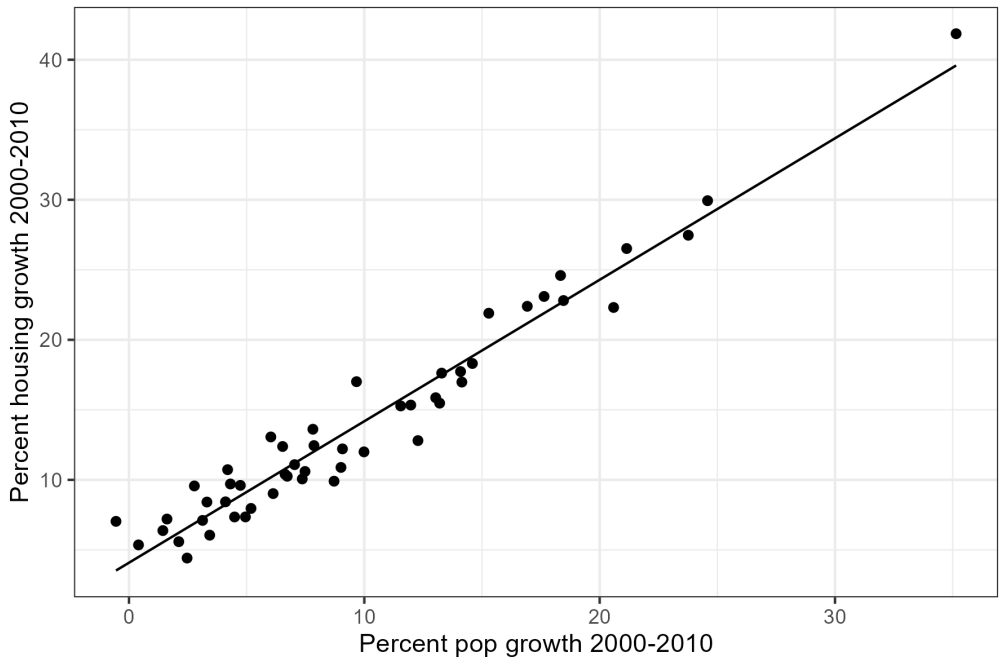
1. **Generalizing from sample to population:** Is this coefficient the same one I would estimate with the entire population?



Building models

What can we do with this model?

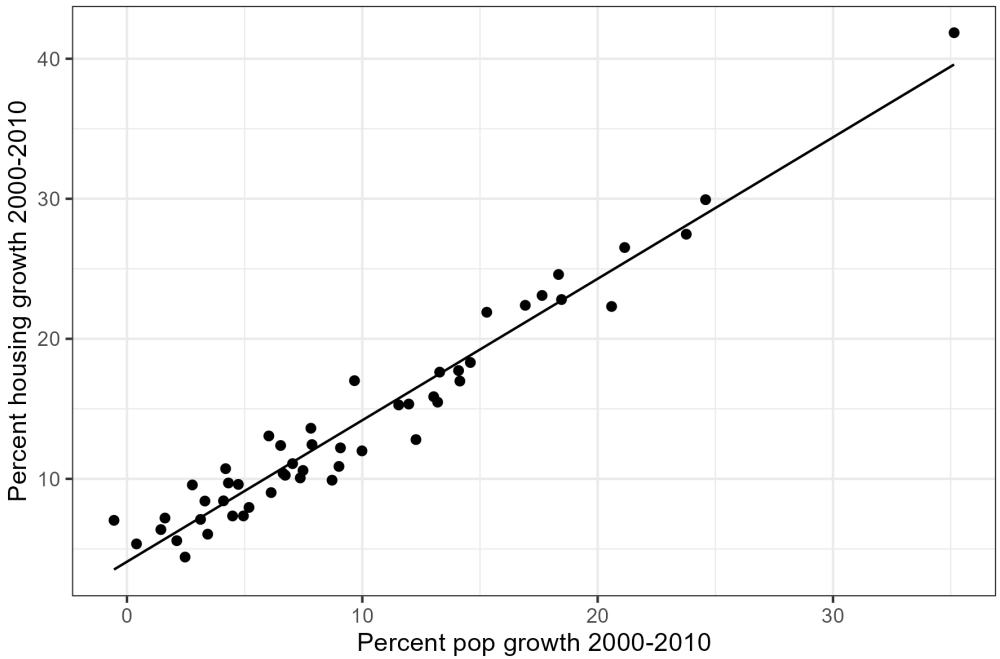
1. Generalizing from sample to population
2. **Measurement:** Can I generalize to all types of housing growth?



Building models

What can we do with this model?

1. Generalizing from sample to population
2. Measurement
3. **Forecasting:** Can I use this model to predict out-of-sample?



Building models

What can we do with this model?

1. Generalizing from sample to population
2. Measurement
3. Forecasting
4. **Causal inference:** Can I say pop growth **causes** housing growth?