

SOC-5811 Week 3: Intro to R and regression

Nick Graetz

University of Minnesota, Department of Sociology

9/17/2025

1/27



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/ncgra/Dropbox/'  
class(dropbox)
```

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

```
length(dropbox)
```

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

```
nchar(dropbox)
```

```
## [1] 23
```


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
```

7/27



UNIVERSITY OF MINNESOTA

Driven to DiscoverSM



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
```

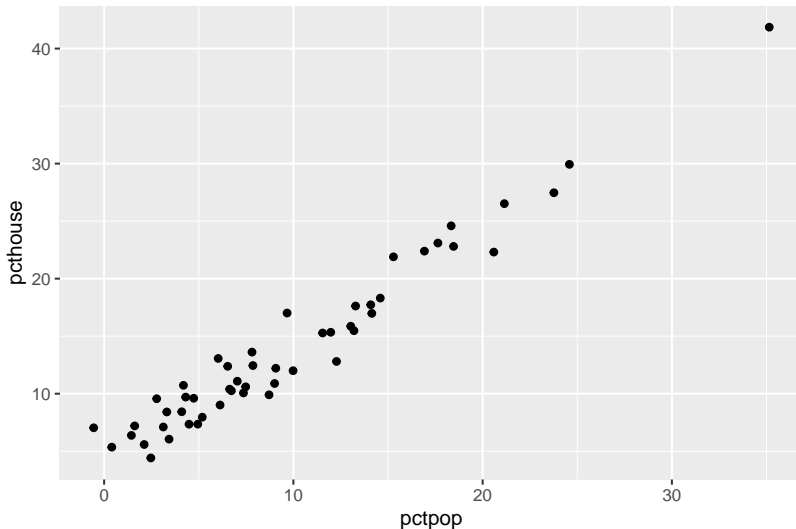
EXAMINE DATA

ggplot and the Grammar of Graphics

<https://ggplot2-book.org/introduction.html>

```
plot <- ggplot (data=census,  
               aes (x=pctpop,  
                   y=pcthouse)) +  
  geom_point ()
```

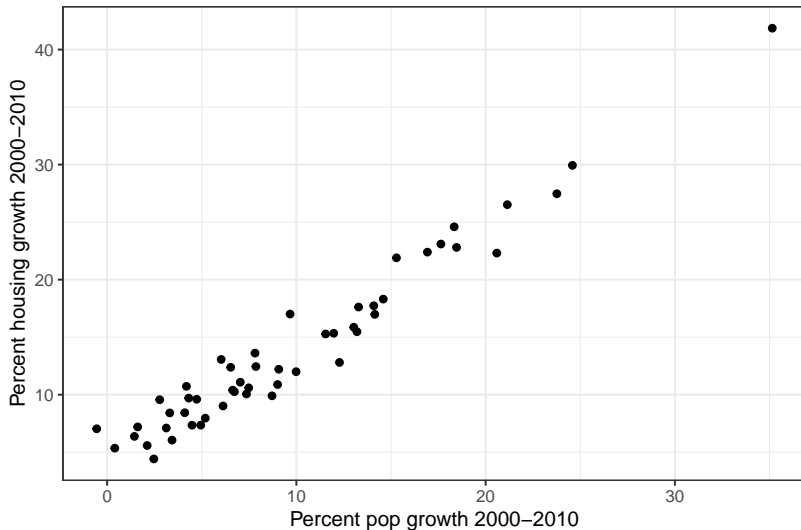
EXAMINE DATA



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()
```

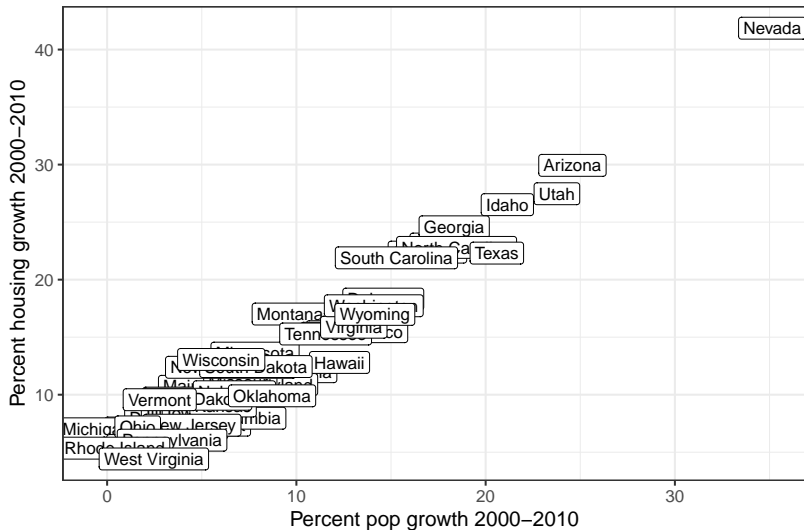
EXAMINE DATA



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:

$$pcthouse = f(pctpop)$$



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
```

18/27

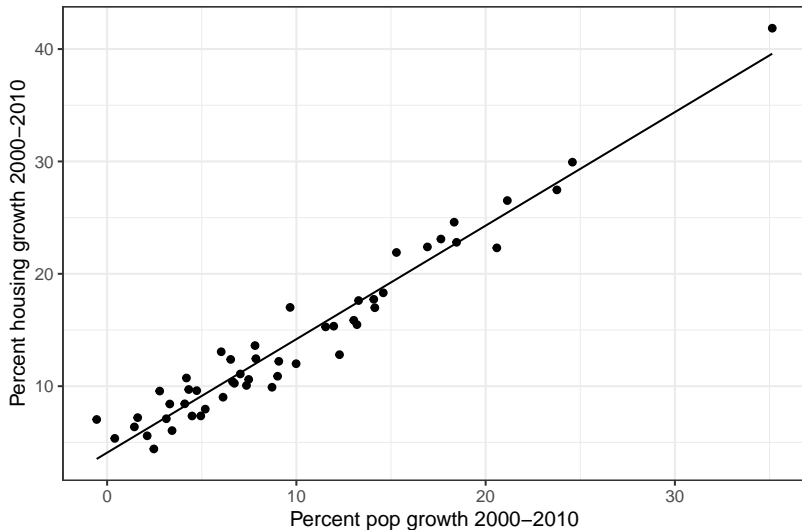


MAKING PREDICTIONS

```
census <- census %>%  
  mutate(pcthouse_pred=predict(model))  
head(census)
```

##	state	statefp	a00aa2000	a00aa2010	a41aa2000	a41aa2010	pctpop
##	<char>	<char>	<num>	<num>	<num>	<num>	<num>
## 1:	Alabama	01	4447100	4779736	1963711	2171853	7.479841
## 2:	Alaska	02	626932	710231	260978	306967	13.286768
## 3:	Arizona	04	5130632	6392017	2189189	2844526	24.585373
## 4:	Arkansas	05	2673400	2915918	1173043	1316299	9.071520
## 5:	California	06	33871648	37253956	12214549	13680081	9.985662
## 6:	Colorado	08	4301261	5029196	1808037	2212898	16.923758
##	pcthouse	onepct	pcthouse_pred				
##	<num>	<num>	<num>				
## 1:	10.59942	0	11.63810				
## 2:	17.62179	1	17.50481				
## 3:	29.93515	2	28.91974				
## 4:	12.21234	3	13.24616				
## 5:	11.99825	4	14.16972				
## 6:	22.39230	5	21.17924				

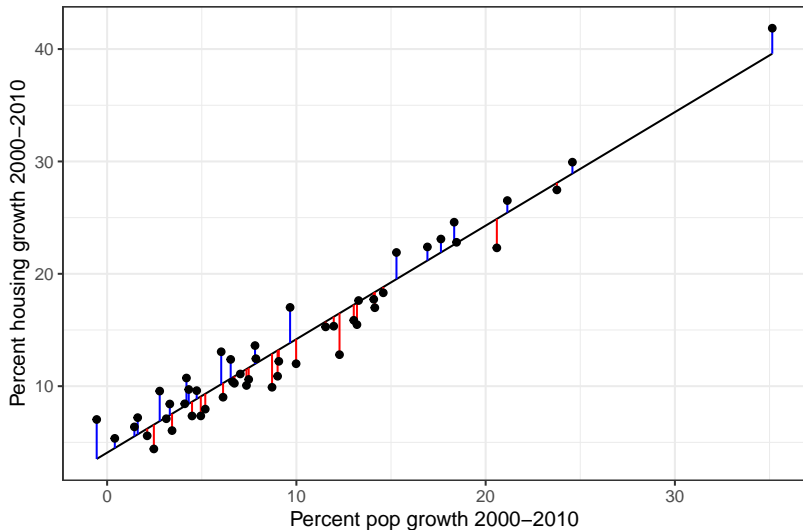
MAKING PREDICTIONS



20/27



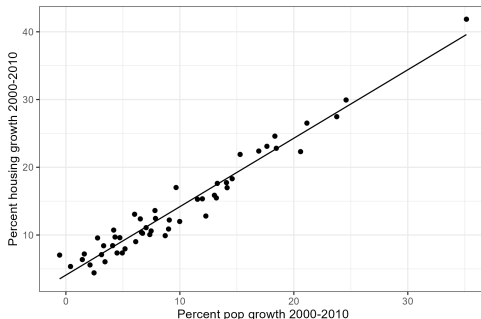
MAKING PREDICTIONS



BUILDING MODELS

What can we do with this model?

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

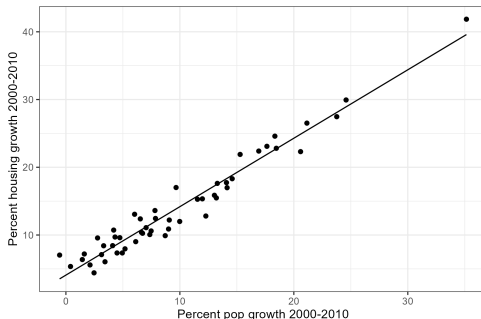


BUILDING MODELS

What can we do with this model?

1. **Generalizing from sample to population:**

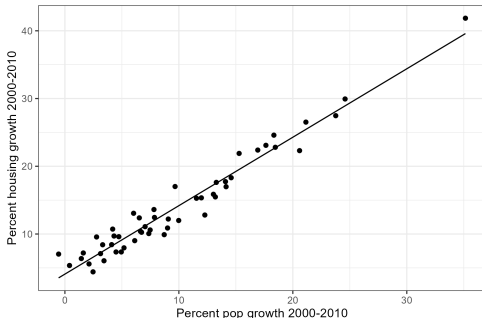
Is this coefficient the same one I would estimate with the entire population?



BUILDING MODELS

What can we do with this model?

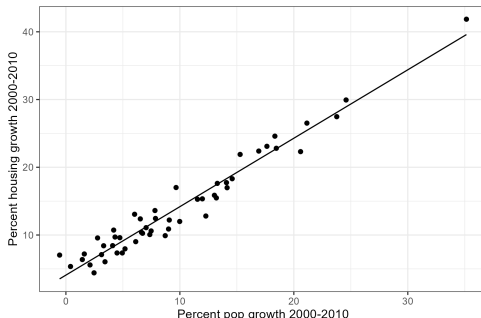
1. Generalizing from sample to population
2. **Measurement:** Can I generalize to all types of housing growth?



BUILDING MODELS

What can we do with this model?

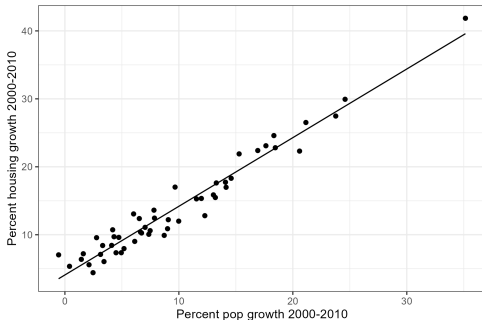
1. Generalizing from sample to population
2. Measurement
3. **Forecasting:** Can I use this model to predict out-of-sample?



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?



BUILDING MODELS

