More regression, model fit, and multiple regression

Seung-Ho An, University of Arizona

Agenda

Model fit

Multiple regression

Categorical independent variables (lab)

More lab!

1. Model fit

Presidential popularity and the midterms

Does popularity of the president or recent changes in the economy better predict midterm election outcomes?

Variable	Description
year	midterm election year
president	name of president
party	Democrat or Republican
approval	Gallup approval rating at midterms
rdi_change	% change in real disposable income over the year before midterms
seat_change	change in the number of House seats for the president's party

```
library(TPDdata)
midterms
```

```
## # A tibble: 20 x 6
##     year president  party approval seat_change rdi_change
##    <dbl> <chr>      <chr>    <dbl>       <dbl>      <dbl>
##  1  1946 Truman     D           33         -55     NA
##  2  1950 Truman     D           39         -29      8.2
##  3  1954 Eisenhower R           61          -4      1
##  4  1958 Eisenhower R           57         -47      1.1
##  5  1962 Kennedy    D           61          -4      5
##  6  1966 Johnson    D           44         -47      5.3
##  7  1970 Nixon      R           58          -8      6.6
##  8  1974 Ford       R           54         -43      6.4
##  9  1978 Carter     D           49         -11      7.7
## 10  1982 Reagan     R           42         -28      4.8
## 11  1986 Reagan     R           63          -5      5.1
## 12  1990 H.W. Bush  R           58          -8      5.6
## 13  1994 Clinton    D           46         -53      3.9
## 14  1998 Clinton    D           66           5      5.6
## 15  2002 W. Bush    R           63           6      2.6
## 16  2006 W. Bush    R           38         -30      5.7
## 17  2010 Obama      D           45         -63      3.5
## 18  2014 Obama      D           40         -13      4.6
## 19  2018 Trump      R           38         -42      4.1
## 20  2022 Biden      D           42          NA     -0.003
```

Fitting the approval model

```r
fit.app <- lm(seat_change ~ approval, data = midterms)
fit.app
```

```
##
## Call:
## lm(formula = seat_change ~ approval, data = midterms)
##
## Coefficients:
## (Intercept)     approval
##      -96.58         1.42
```

Let's write out the mathematical model together and then interpret the coefficient!

For a one-point increase in presidential approval, the predicted seat change increases by 1.42

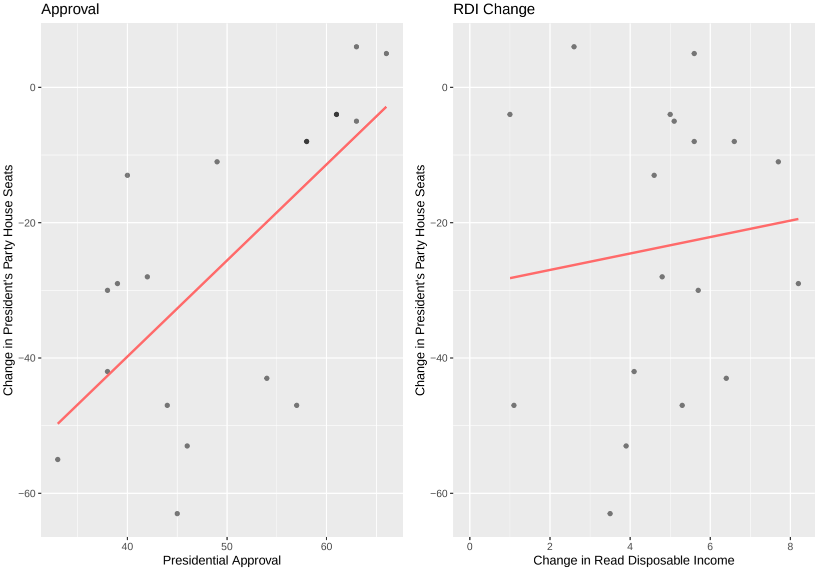
Fitting the income model

Now look at the change in real disposable income as an independent variable

```
fit.rdi <- lm(seat_change ~ rdi_change, data = midterms)
fit.rdi
```

```
##
## Call:
## lm(formula = seat_change ~ rdi_change, data = midterms)
##
## Coefficients:
## (Intercept)   rdi_change
##     -29.413        1.215
```

For a one point increase in the change in real disposable income, the predicted seat change increases by 1.21.



How well do the models “fit the data”?

- How well does the model predict the outcome variable in the data?

Model fit

Model prediction error:

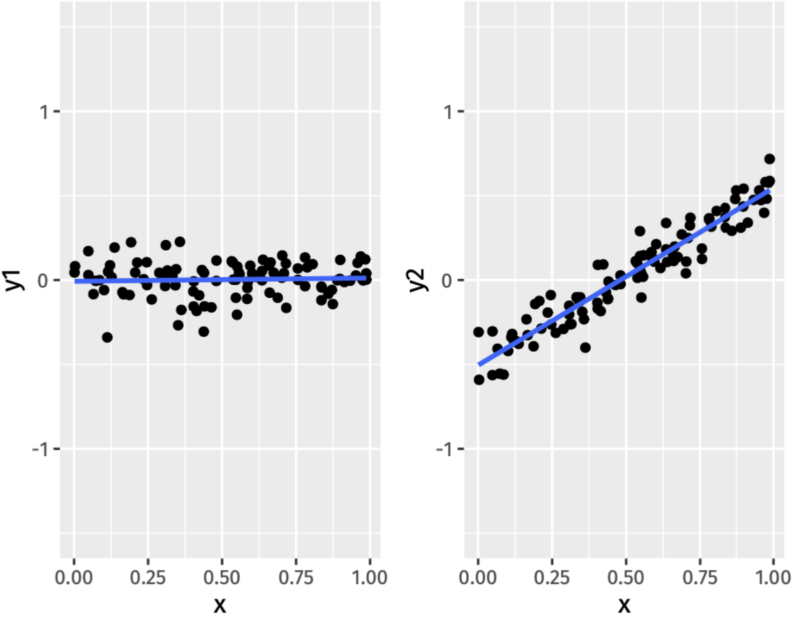
$$\text{prediction error} = \sum_{i=1}^{n} (\text{actual}_i - \text{predicted}_i)^2$$

Prediction error for regression: **Sum of squared residuals**

$$SSR = \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2$$

Lower SSR is better, right?

These two regression lines have approximately the same SSR:



Benchmarking model fit

Benchmarking our prediction using the **proportional reduction in error:**

$$\frac{\text{reduction in prediction error using model}}{\text{baseline prediction error}}$$

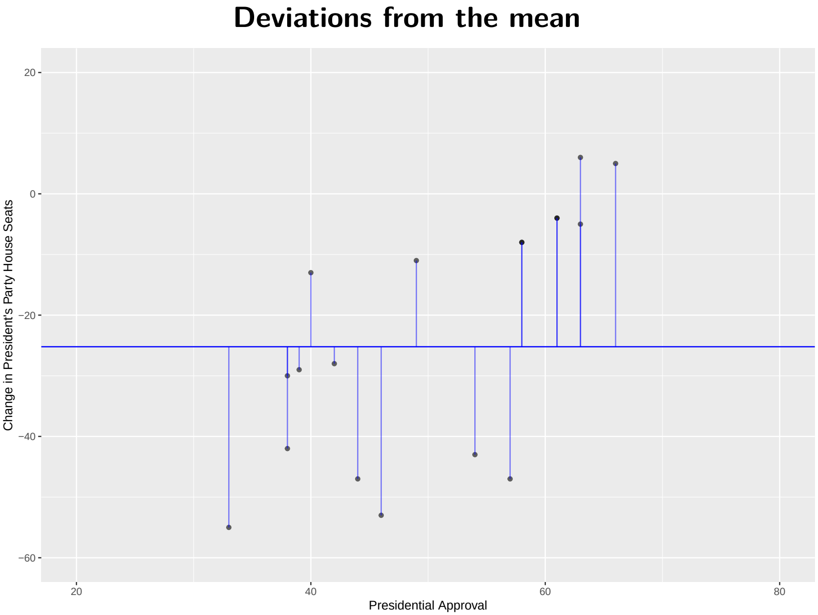
Baseline prediction error without a regression is using the mean of `Y` to predict. This is called the **total sum of squares**:

$$TSS = \sum_{i=1}^{n}(Y_i - \overline{Y})^2$$

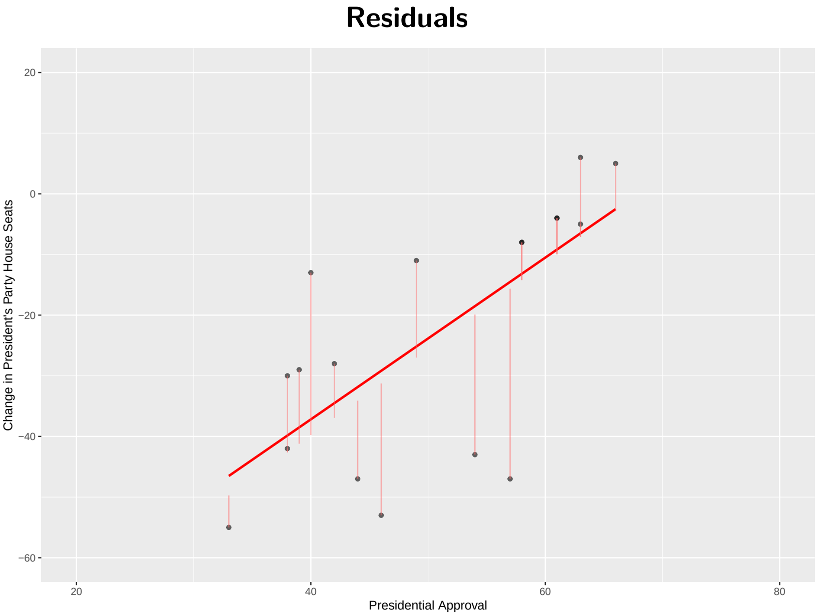
Leads to the **coefficient of determination**, R^2, one summary of LS model fit:

$$R^2 = \frac{TSS - SSR}{TSS} = \frac{\text{how much smaller LS prediction errors are vs mean}}{\text{prediction error using the mean}}$$

Total SS vs. SSR



Total SS vs. SSR



Model fit in R

- To access R^2 from the lm() output, use the summary() function:

```
fit.app.sum <- summary(fit.app)
fit.app.sum$r.squared
```

```
## [1] 0.4498696
```

- Compare to the fit using change in income:

```
fit.rdi.sum <- summary(fit.rdi)
fit.rdi.sum$r.squared
```

```
## [1] 0.01202348
```

Which does a better job predicting midterm election outcomes?

Accessing model fit via broom package

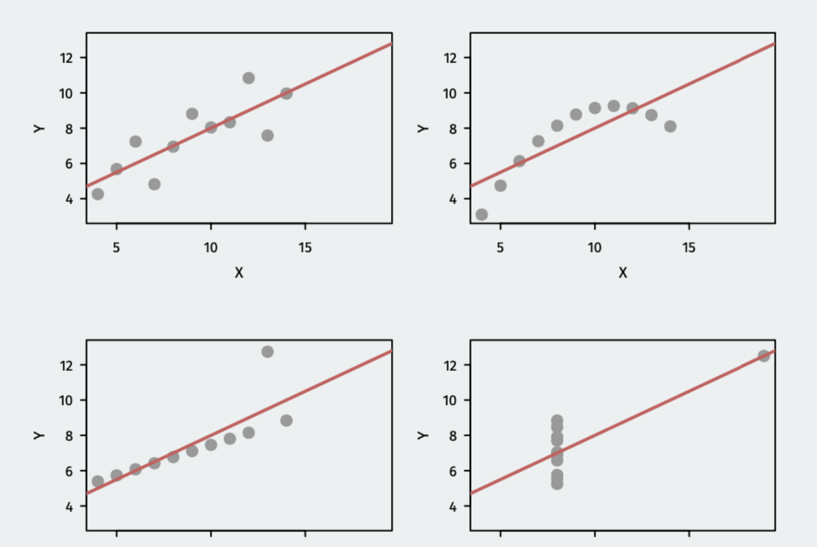
We can also access summary statistics like model fit using the glance() function from broom:

```
library(broom)
glance(fit.app)
```

```
## # A tibble: 1 x 12
##   r.squ~1 adj.r~2 sigma stati~3 p.value    df logLik   AIC   BIC devia~4 df.re~5
##     <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>   <dbl> <dbl> <dbl>   <dbl>   <int>
## 1   0.450   0.418  16.9    13.9 0.00167     1  -79.6  165.  168.   4852.      17
## # ... with 1 more variable: nobs <int>, and abbreviated variable names
## #   1: r.squared, 2: adj.r.squared, 3: statistic, 4: deviance, 5: df.residual
```

Is R-squared useful?

- Can be very misleading. Each of these samples have the same R^2 even though they are vastly different:



Overfitting

- **In-sample fit**: how well your model predicts the data used to estimate it
- R^2 is a measure of in-sample fit
- **Out-of-sample fit**: how well your model predicts new data
- **Overfitting**: OLS optimizes in-sample fit; may do poorly out of sample
    - Example: predicting winner of Democratic presidential primary with gender of the candidate
    - Until 2016, gender was a perfect predictor of who wins the primary
    - Prediction for 2016 based on this: Bernie Sanders as Dem. nominee
    - Bad out-of-sample prediction due to overfitting

2. Multiple regression

Multiple predictors

What if we want to predict Y as a function of many variables?

$$\text{seat\_change}_i = \alpha + \beta_1 \textit{approval}_i + \beta_2 \text{rdi\_change}_i + \epsilon_i$$

Why?

- Better predictions (at least in-sample)
- Better interpretation as **ceteris paribus** relationships:
  - β_1 is the relationship between `approval` and `seat_change` holding `rdi_change` constant
  - **Statistical control** in a cross-sectional study

Multiple regression in R

```r
mult.fit <- lm(seat_change ~ approval + rdi_change, data = midterms)
mult.fit
```

```
##
## Call:
## lm(formula = seat_change ~ approval + rdi_change, data = midterms)
##
## Coefficients:
## (Intercept)     approval   rdi_change
##    -117.226        1.526        3.217
```

- $\widehat{\alpha} = -117.2$: average seat change president has 0% approval and no change in income levels
- $\widehat{\beta_1} = 1.53$ average increase in seat change for additional percentage point of approval, **holding RDI change fixed**
- $\widehat{\beta_2} = 3.217$: average increase in seat for each additional percentage point increase of RDI, **holding approval fixed**

Least squares with multiple regression

- How do we estimate the coefficients?
- The same exact way as before: minimize prediction error!
- Residuals (aka prediction error) with multiple predictors:

$$Y_i - \widehat{Y}_i = \text{seat\_change}_i - \widehat{\alpha} - \widehat{\beta}_1 \text{approval}_i - \widehat{\beta}_2 \text{rdi\_change}_i$$

- Find the coefficients that minimizes the **sum of the squared residuals:**

$$SSR = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = (Y_i - \widehat{\alpha} - \widehat{\beta}_1 X_{i,1} - \widehat{\beta}_2 X_{i,2})^2$$

Model fit with multiple predictors

- R^2 mechanically increases when you add a variable to the regression
- But this could be overfitting!!
- Solution: penalize regression models with more variables
- Occam's razor: **simpler models are preferred**
- Adjusted R^2: lowers regular R^2 for each additional covariate
- If the added covariate doesn't help predict, adjusted R^2 goes down

Comparing model fits

```
glance(fit.app) |>
  select(r.squared, adj.r.squared)
```

```
## # A tibble: 1 x 2
##   r.squared adj.r.squared
##       <dbl>         <dbl>
## 1     0.450         0.418
```

```
glance(mult.fit) |>
  select(r.squared, adj.r.squared)
```

```
## # A tibble: 1 x 2
##   r.squared adj.r.squared
##       <dbl>         <dbl>
## 1     0.468         0.397
```

Predicted values from R

We could plug in values into the equation, but R can do this for us. The {modelr} package gives some functions that allow us to predictions in a tidy way:

Let's use add_predictions() to predict the 2022 results

```
library(modelr)
```

```
##
## Attaching package: 'modelr'

## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
midterms |>
  filter(year == 2022) |>
  add_predictions(mult.fit)
```

```
## # A tibble: 1 x 7
##    year president party approval seat_change rdi_change  pred
##   <dbl> <chr>     <chr>    <dbl>       <dbl>      <dbl> <dbl>
## 1  2022 Biden     D           42          NA     -0.003 -53.2
```

Predictions from several models

The gather_predictions() will return one row for each model passed to it with the prediction for that model:

```
midterms |>
  filter(year == 2022) |>
  gather_predictions(fit.app, mult.fit)
```

```
## # A tibble: 2 x 8
##   model     year president party approval seat_change rdi_change  pred
##   <chr>    <dbl> <chr>     <chr>    <dbl>       <dbl>      <dbl> <dbl>
## 1 fit.app   2022 Biden     D           42          NA     -0.003 -36.9
## 2 mult.fit  2022 Biden     D           42          NA     -0.003 -53.2
```

Predictions from new data

What about predicted values not in data?

```
tibble(approval = c(50, 75), rdi_change = 0) |>
  gather_predictions(fit.app, mult.fit)
```

```
## # A tibble: 4 x 4
##   model    approval rdi_change   pred
##   <chr>       <dbl>      <dbl>  <dbl>
## 1 fit.app        50          0 -25.6
## 2 fit.app        75          0   9.92
## 3 mult.fit       50          0 -40.9
## 4 mult.fit       75          0  -2.79
```

Predictions from augment()

We can also get predicted values from the augment() function using the newdata argument:

```r
newdata <- tibble(approval = c(50, 75), rdi_change = 0)

augment(mult.fit, newdata = newdata)
```

```
## # A tibble: 2 x 3
##   approval rdi_change .fitted
##      <dbl>      <dbl>   <dbl>
## 1       50          0  -40.9
## 2       75          0   -2.79
```

3. Categorical independent variables (lab session)

Political effects of gov't programs

- *progesa*: Mexican conditional cash transfer program (CCT) from $\sim$2000
- Welfare $$ given if kids enrolled in schools, get regular check-ups, etc.
- Do these programs have political effects?
- Program had support from most parties
- Was implemented in a nonpartisan fashion
- Would the incumbent presidential party be rewarded?

The data

- Randomized roll-out of the CCT program:
- treatment: receive CCT 21 months before 2000 election
- control: receive CCT 6 months before 2000 election
- Does having CCT longer mobilize voters for incumbent PRI party?

Name	Description
treatment	early Progresa (1) or late Progresa (0)
pri2000s	PRI votes in the 2000 election as a share of adults in precinct
t2000	turnout in the 2000 election as share of adults in precinct

```r
library(qss)
data("progresa", package = "qss")
cct <- as_tibble(progresa) |>
  select(treatment, pri2000s, t2000)
cct
```

```
## # A tibble: 417 x 3
##    treatment pri2000s t2000
##        <int>    <dbl> <dbl>
##  1         1     40.8  55.8
##  2         1     22.4  31.2
##  3         1     38.9  47.0
##  4         1     31.2  45.0
##  5         0     76.9 100
##  6         0     23.9  37.4
##  7         1     47.3  64.9
##  8         1     21.4  58.1
##  9         1     56.5  71.3
## 10         1     36.6  51.2
## # ... with 407 more rows
```

Difference in means estimates

Let's calculate difference in means (ATEs!!) for the follow two questions.

Does CCT affect turnout?

Does CCT affect PRI (incumbent) votes?

You will be likely to need `group_by`, `summarize`, `pivot_wider`, and `mutate` functions. Also, I promise this will be the last time that you will be performing differences in means in this class! If you don't recall how to do so, consult with the topic of causality in week 3

Binary independent variables (lecture)

$$Y_i = \alpha + \beta X_i \epsilon_i$$

- When independent variable X_i is **binary:**
- Intercept $\widehat{\alpha}$ is the average outcome in the X $=$ 0 group
- Slope $\widehat{\beta}$ is the difference-in-means of Y between X $=$ 1 group and X $=$ 0 group

$$\widehat{\beta} = \overline{Y}_{treated} - \overline{Y}_{control}$$

- If there are other independent variables, this becomes the difference-in-means controlling for those covariates

Linear regression for experiments (exercise)

- Under **randomization**, we can estimate the ATE with regression.

Let's run a regression model to calculate the effects of CCT on PRI (incumbent) votes.

Wait a second..

Categorical variables in regression (lecture)

- We often have **categorical variables:**
  - Race/ethnicity: white, Black, Latino, Asian
  - Partisanship: Democrat, Republican, Independent
- Strategy for including in a regression: create a **series of binary variables**

unit	Party	Democrat	Republican	Independent
1	Democrat	1	0	0
2	Democrat	1	0	0
3	Independent	0	0	1
4	Republican	0	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

- Then include **all but one** of these binary variables:

$$turnout_i = \alpha + \beta_1 \text{Republican}_i + \beta_2 \text{Independent}_i + \epsilon_i$$

Interpreting categorical variables

$$\textit{turnout}_i = \alpha + \beta_1 \text{Republican}_i + \beta_2 \text{Independent}_i + \epsilon_i$$

- $\widehat{\alpha}$: average outcome in the **omitted group/baseline** (Democrats)
- $\widehat{\beta}$: average difference between each group and the baseline
  - $\widehat{\beta}_1$: average difference in turnout between Republicans and Democrats
  - $\widehat{\beta}_2$: average difference in turnout between Independents and Democrats

More lab exercise: the transphobia study

```r
transphobia<-read_csv("data/transphobia_all.csv")
```

```
## Rows: 9110 Columns: 11
## -- Column specification -------------------------------
## Delimiter: ","
## chr (3): wave, treat_ind, racename
## dbl (8): id, age, female, voted_gen_14, voted_gen_12, de
##
## i Use `spec()` to retrieve the full column specificatio
## i Specify the column types or set `show_col_types = FALS
```

Q1

Run a regression model of thermometer scores for transgender people in wave 0 (baseline) on the treatment indicator (treat_ind) and gender. Store the regression outputs into a vector (or a variable) and then use the `summary` function to call the vector (or the variable).

Q2

Run a regression of thermometer scores for transgender people in wave 1 (3 days) on the treatment indicator (treat_ind), political ID (democrat), and gender (female).

Interpret the coefficients of the two variables.

Q3

Let's run a regression model of thermometer scores for transgender on age, female, and democrat. What is the predicted value of thermometer scores for transgender for 50 years old republican female? (you can use R to do so or simply plugging the numbers into the equation) What is the R squared? How would you interpret it?