Gov 50: 15. Multiple Regression and Interpretation

Matthew Blackwell

Harvard University

Roadmap

1. Multiple regression
2. Categorical independent variables

1/ Multiple regression

Multiple predictors

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

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

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

```
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
##     -117.23         1.53         3.22
```

- $\hat{\alpha} =$ -117.2: average seat change president has 0% approval and no change in income levels.
- $\hat{\beta}_1 = 1.53$: average increase in seat change for additional percentage point of approval, **holding RDI change fixed**
- $\hat{\beta}_2 = 3.217$: average increase in seat change 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:

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

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

$$\text{SSR} = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = (Y_i - \hat{\alpha} - \hat{\beta}_1 X_{i1} - \hat{\beta}_2 X_{i2})^2$$

Model fit with multiple predictors

- R^2 mechanically increases when you add a variables 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 covariates doesn't help predict, adjusted R^2 goes down!

Comparing model fits

```r
library(broom)
fit.app <- lm(seat_change ~ approval, data = midterms)
glance(fit.app) |>
  select(r.squared, adj.r.squared, sigma)
```

```
## # A tibble: 1 x 3
##   r.squared adj.r.squared sigma
##       <dbl>         <dbl> <dbl>
## 1     0.450         0.418  16.9
```

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

```
## # A tibble: 1 x 3
##   r.squared adj.r.squared sigma
##       <dbl>         <dbl> <dbl>
## 1     0.468         0.397  16.7
```

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)

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

```
## # A tibble: 1 x 7
##    year president party approval seat_change rdi_cha~1  pred
##   <dbl> <chr>     <chr>    <dbl>       <dbl>     <dbl> <dbl>
## 1  2022 Biden     D           42          NA    -0.003 -53.2
## # ... with abbreviated variable name 1: rdi_change
```

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 presi~1 party appro~2 seat_~3 rdi_c~4  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
## # ... with abbreviated variable names 1: president,
## #   2: approval, 3: seat_change, 4: rdi_change
```

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
```

2/ Categorical independent variables

Political effects of gov't programs

- *Progesa*: Mexican conditional cash transfer program (CCT) from ~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

Does CCT affect turnout?

```
cct |> group_by(treatment) |>
  summarize(t2000 = mean(t2000)) |>
  pivot_wider(names_from = treatment, values_from = t2000) |>
  mutate(ATE = `1` - `0`)
```

```
## # A tibble: 1 x 3
##     `0`   `1`   ATE
##   <dbl> <dbl> <dbl>
## 1  63.8  68.1  4.27
```

Does CCT affect PRI (incumbent) votes?

```
cct |> group_by(treatment) |>
  summarize(pri2000s = mean(pri2000s)) |>
  pivot_wider(names_from = treatment, values_from = pri2000s) |>
  mutate(ATE = `1` - `0`)
```

```
## # A tibble: 1 x 3
##     `0`   `1`   ATE
##   <dbl> <dbl> <dbl>
## 1  34.5  38.1  3.62
```

Binary independent variables

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

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

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

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

Linear regression for experiments

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

```
cct |> group_by(treatment) |>
  summarize(pri2000s = mean(pri2000s)) |>
  pivot_wider(names_from = treatment, values_from = pri2000s) |>
  mutate(ATE = `1` - `0`)
```

```
## # A tibble: 1 x 3
##     `0`   `1`   ATE
##   <dbl> <dbl> <dbl>
## 1  34.5  38.1  3.62
```

```
lm(pri2000s ~ treatment, data = cct) |> coef()
```

```
## (Intercept)   treatment
##       34.49        3.62
```

Categorical variables in regression

- 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
⋮	⋮	⋮	⋮	⋮

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

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

Interpreting categorical variables

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

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

CCES data

```
library(gov50data)
cces_2020
```

```
## # A tibble: 51,551 x 6
##    gender race  educ                 pid3    turno~1 pres_~2
##    <fct>  <fct> <fct>                <fct>     <dbl> <fct>
##  1 Male   White 2-year               Republ~       1 Donald~
##  2 Female White Post-grad            Democr~      NA <NA>
##  3 Female White 4-year               Indepe~       1 Joe Bi~
##  4 Female White 4-year               Democr~       1 Joe Bi~
##  5 Male   White 4-year               Indepe~       1 Other
##  6 Male   White Some college         Republ~       1 Donald~
##  7 Male   Black Some college         Not su~      NA <NA>
##  8 Female White Some college         Indepe~       1 Donald~
##  9 Female White High school graduate Republ~       1 Donald~
## 10 Female White 4-year               Democr~       1 Joe Bi~
## # ... with 51,541 more rows, and abbreviated variable names
## #   1: turnout_self, 2: pres_vote
```

Categorical variables in the CCES data

```r
turnout_pred <- lm(turnout_self ~ pid3, data = cces_2020)
turnout_pred
```

```
##
## Call:
## lm(formula = turnout_self ~ pid3, data = cces_2020)
##
## Coefficients:
##     (Intercept)   pid3Republican  pid3Independent
##          0.9635          -0.0103          -0.0394
##       pid3Other     pid3Not sure
##         -0.0066          -0.3331
```

What R does internally with factor variables in `lm`

```r
cces_2020 |> drop_na(turnout_self, pid3) |> select(pid3) |> pull() |>
  head()
```

```
## [1] Republican  Independent Democrat    Independent
## [5] Republican  Independent
## 7 Levels: Democrat Republican Independent ... not asked
```

```r
model.matrix(turnout_pred) |>
  head()
```

```
##   (Intercept) pid3Republican pid3Independent pid3Other
## 1           1              1               0         0
## 3           1              0               1         0
## 4           1              0               0         0
## 5           1              0               1         0
## 6           1              1               0         0
## 8           1              0               1         0
##   pid3Not sure
## 1            0
## 3            0
## 4            0
## 5            0
## 6            0
```