Multiple Regression

Dale Barr
University of Glasgow

Moving beyond simple regression

- dealing with multiple predictors
- model comparison
- coding categorical predictors

Dealing with multiple predictors

Multiple regression

General model for single-level data with m predictors:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_m X_{mi} + e_i$$

individual Xs can be any combination of continuous and categorical predictors (and their interactions)

Each β_j is the *partial effect of X_j holding all other Xs constant*

NB: single-level data is rare in psychology

Example

Are lecture attendance and engagement with online materials associated with higher grades in statistics?

Does this relationship hold after controlling for overall GPA?

Import

grades.csv

```r
grades <- read_csv("data/grades.csv", col_types = "ddii")

grades
```

```
# A tibble: 100 × 4
   grade   GPA lecture nclicks
   <dbl> <dbl>   <int>   <int>
 1  2.40 1.13        6      88
 2  3.67 0.971       6      96
 3  2.85 3.34        6     123
 4  1.36 2.76        9      99
 5  2.31 1.02        4      66
 6  2.58 0.841       8      99
 7  2.69 4           5      86
 8  3.05 2.29        7     118
 9  3.21 3.39        9      98
10  2.24 3.27       10     115
# ℹ 90 more rows
```

Correlations

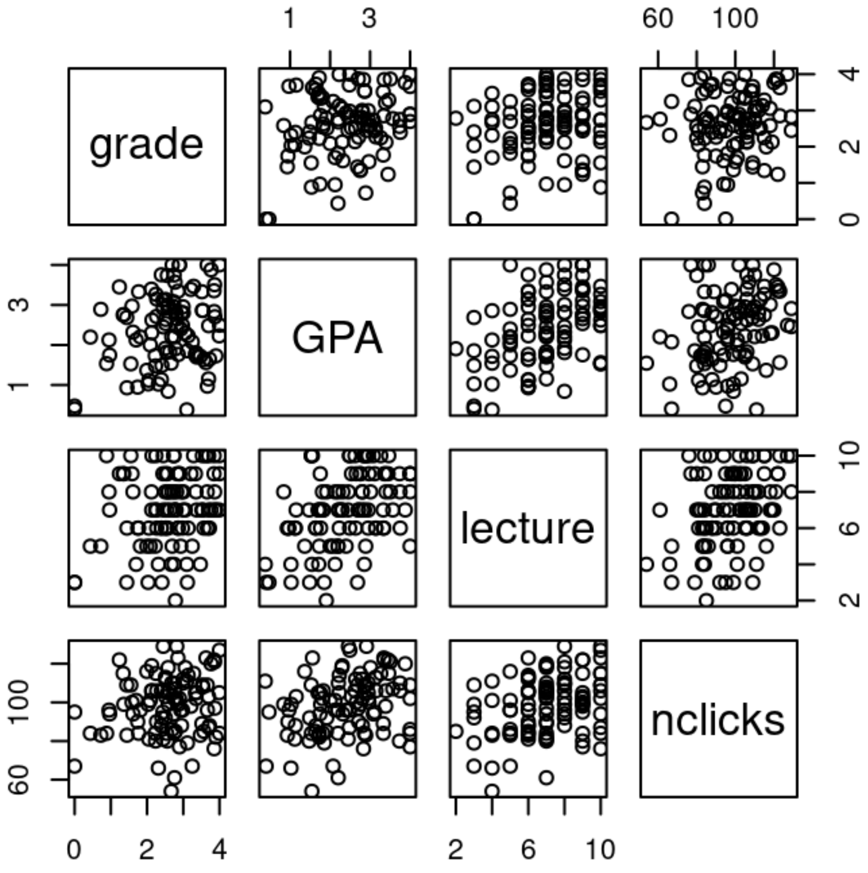
```
library("corrr")

grades %>%
  correlate() %>%
  shave() %>%
  fashion()
```

```
     term grade  GPA lecture nclicks
1   grade
2     GPA   .25
3 lecture   .24  .44
4 nclicks   .16  .30     .36
```

Visualization

```
pairs(grades)
```



Estimation

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_m X_{mi} + e_i$$

`lm(Y ~ X1 + X2 + ... + Xm, data)`

```
my_model <- lm(grade ~ lecture + nclicks, grades)
```

Output

```
summary(my_model)
```

```
Call:
lm(formula = grade ~ lecture + nclicks, data = grades)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.21653 -0.40603  0.02267  0.60720  1.38558 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 1.462037   0.571124   2.560   0.0120 *
lecture     0.091501   0.045766   1.999   0.0484 *
nclicks     0.005052   0.006051   0.835   0.4058  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Standardized coefficients

```
grades2 <- grades %>%
  mutate(lecture_c = (lecture - mean(lecture)) / sd(lecture),
         nclicks_c = (nclicks - mean(nclicks)) / sd(nclicks))

summary(lm(grade ~ lecture_c + nclicks_c, grades2))
```

Standardized coefficients

```
Call:
lm(formula = grade ~ lecture_c + nclicks_c, data = grades2)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.21653 -0.40603  0.02267  0.60720  1.38558 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.59839    0.08692  29.895   <2e-16 ***
lecture_c    0.18734    0.09370   1.999   0.0484 *  
nclicks_c    0.07823    0.09370   0.835   0.4058    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model comparison

Model comparison

Is engagement (as measured by lecture attendance and downloads) positively associated with final course grade *above and beyond* student ability (as measured by GPA)?

Strategy

Compare "base" model with control vars to a "bigger" model with control plus focal vars

```
base_model <- lm(grade ~ GPA, grades)
big_model <- lm(grade ~ GPA + lecture + nclicks, grades)

anova(base_model, big_model)
```

```
Analysis of Variance Table

Model 1: grade ~ GPA
Model 2: grade ~ GPA + lecture + nclicks
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     98 73.528
2     96 71.578  2    1.9499 1.3076 0.2752
```

$$F(2, 96) = 1.31, p = .275$$

If $p < \alpha$, bigger model is better.

update()

```r
base_model <- lm(grade ~ GPA, grades)
big_model <- update(base_model, . ~ . +lecture +nclicks)

anova(base_model, big_model)
```

```
Analysis of Variance Table

Model 1: grade ~ GPA
Model 2: grade ~ GPA + lecture + nclicks
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     98 73.528
2     96 71.578  2    1.9499 1.3076 0.2752
```

Categorical predictors

Dummy coding binary variables

$k = 2$

Arbitrarily assign one levels to 0; assign the other to 1.

```
dat |>
  mutate(dummy = if_else(predictor == "targetlevel", 1, 0))
```

See `dplyr::if_else()`

NB: sign of the variable depends on the coding!

Factors with $k > 2$

Arbitrarily choose one level as “baseline” level. Need $k - 1$ predictors, each contrasting a target level with baseline.

$k = 3$

	A2v1	A3v1
A_1	0	0
A_2	1	0
A_3	0	1

$k = 4$

	A2v1	A3v1	A4v1
A_1	0	0	0
A_2	1	0	0
A_3	0	1	0
A_4	0	0	1

Bodyweight over the seasons

```
set.seed(1451)

season_wt <- tibble(season = rep(c("winter", "spring", "summer", "fall"),
                                 each = 5),
                    bodyweight_kg = c(rnorm(5, 105, 3),
                                      rnorm(5, 103, 3),
                                      rnorm(5, 101, 3),
                                      rnorm(5, 102.5, 3)))

season_wt
```

```
# A tibble: 20 × 2
   season bodyweight_kg
   <chr>          <dbl>
 1 winter          96.9
 2 winter         102. 
 3 winter         101. 
 4 winter         107. 
 5 winter         106. 
 6 spring         109. 
 7 spring         103. 
 8 spring          99.9
 9 spring          98.5
10 spring         103. 
```

```
11 summer            108.
12 summer            104.
```

Coding the predictor

```
## baseline value is 'winter'
season_wt2 <- season_wt %>%
  mutate(spring_v_winter = if_else(season == "spring", 1, 0),
         summer_v_winter = if_else(season == "summer", 1, 0),
         fall_v_winter = if_else(season == "fall", 1, 0))

## ALWAYS double check using 'distinct'
season_wt2 |>
  distinct(season, spring_v_winter, summer_v_winter, fall_v_winter)
```

```
# A tibble: 4 × 4
  season spring_v_winter summer_v_winter fall_v_winter
  <chr>            <dbl>           <dbl>         <dbl>
1 winter               0               0             0
2 spring               1               0             0
3 summer               0               1             0
4 fall                 0               0             1
```

Fitting the model

```r
mod <- lm(bodyweight_kg ~ spring_v_winter +
            summer_v_winter + fall_v_winter,
          season_wt2)

summary(mod)
```

```
Call:
lm(formula = bodyweight_kg ~ spring_v_winter + summer_v_winter +
    fall_v_winter, data = season_wt2)

Residuals:
    Min      1Q  Median      3Q     Max
-5.7058 -2.1083 -0.5378  1.2883  7.5928

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     102.57645    1.71619  59.770   <2e-16 ***
spring_v_winter  -0.03665    2.42705  -0.015    0.988
summer_v_winter   1.02200    2.42705   0.421    0.679
fall_v_winter    -0.98818    2.42705  -0.407    0.689
```

Main effect of season?

```
mod_base <- lm(bodyweight_kg ~ 1, season_wt2)

anova(mod_base, mod)
```

```
Analysis of Variance Table

Model 1: bodyweight_kg ~ 1
Model 2: bodyweight_kg ~ spring_v_winter + summer_v_winter + fall_v_winter
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     19 245.74
2     16 235.62  3    10.112 0.2289 0.8749
```

One-way Analysis of Variance

```r
season_wt3 <- season_wt2 %>%
  mutate(season = factor(season, levels = c("winter", "spring",
                                            "summer", "fall")))

my_anova <- aov(bodyweight_kg ~ season, season_wt3)
summary(my_anova)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
season       3  10.11   3.371   0.229  0.875
Residuals   16 235.62  14.726
```