Lecture 8: Multiple Regression

C91AR: Advanced Statistics using R

Dr Peter E McKenna

2025-03-11

Contents

1 Setup code

```r
# change output format
options(scipen = 999)

# set the seed
set.seed(453)
```

```
# load packages
pacman::p_load(corrr,
               tidyverse,
               psych,
               tidyplots)
```

2 Content for today

- Multiple regression formula
- Worked example using the "grades.csv" dataset from PsyTeachR
- The `predict` function
- Partial effects
- Standardising coefficients
- Model comparison

3 Getting started with Multiple regression

The general model for single-level data with m predictors is:

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

- Key assumption is that the model residuals are normally distributed.
- Predictor variables X can be either categorical or continuous, as well as interactions between predictors.
- e_i = difference between the predicted and the observed value of Y for the ith participant.
- The relationship is planar, i.e., can be described by a flat surface.
- Error variable is independent of the predictor values.

4 Coefficients

- In multiple regression you will have $m+1$ regression coefficients; one for the intercept (β_0), and one for each predictor (X_m).
- Each β_h value (coefficient associated with the h^{th} independent variable) is understood as the partial effect of β_h holding constant all other predictors.

- In other words, a partial effect of a coefficient in multiple regression refers to the effect of a particular IV on the DV, whilst holding all other IVs constant.
- Response variable (Y) is predicted from a combination of all of the variables multiplied by their respective coefficients, plus a residual term.

5 What is the purpose of multiple regression?

- **To identify a linear combination of predictors that exhibits the highest correlation with the response variable.**

6 A worked example using the `grades.csv` dataset

- How do you get a good grade in statistics?

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

grades
```

```
## # A tibble: 100 x 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
```

6.1 Metadata

- N=100 statistics students

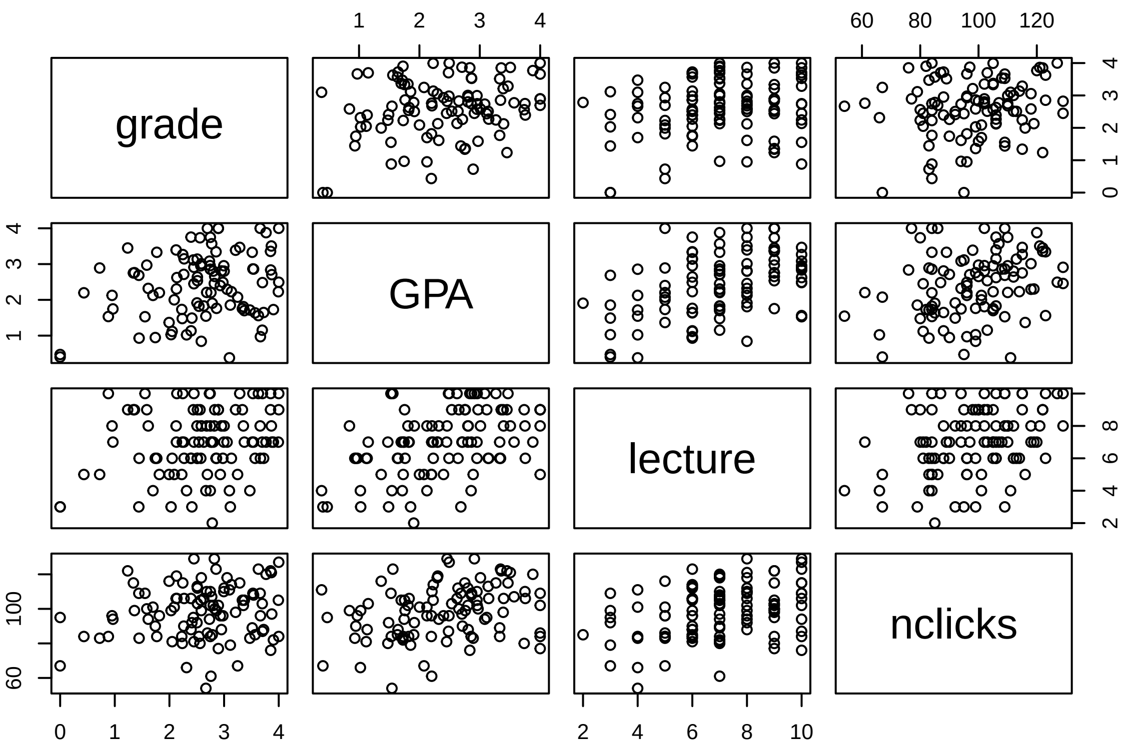
- `grade` = final course grade
- `lecture` = number of lectures attended; an integer from 0:10
- `nclicks` = number of times the students clicked to download online materials
- `GPA` = grade point average prior to taking the course; ranging from 0 (fail) to 4 (best possible grade)

6.2 Examine pairwise correlations

```
# Examine pairwise correlations
grades |>
  correlate() |>
  shave() |>
  fashion() # shave & fashion tidy up the output
```

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

```
pairs(grades)
```



What can you infer from the correlation matrix?

7 Estimation and interpretation

- For a Generalised Linear Model (GLM) with m predictors:

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

- Where...
  - Y_i = the *response variable* (or, the outcome to be predicted)
  - β_0 = the intercept term
  - $\beta_1 X_{1i}$ = the regression coefficient for predictor variable X_1
  - $\beta_2 X_{2i}$ = the regression coefficient for predictor variable X_2
  - e_i = model residuals
  - $h\hat{a}t$ = presence of a hat denotes and sample estimate, not the actual sample statistic

8 Writing out the formula in R

- Writing out a multiple regression model in R is much like what we did for simple regression, except you need to add a term for each predictor variable (X):

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

- **Note**: You do not need to specify the intercept or the residuals, as these are included by default.

9 Predicting `grade` based on `lecture` and `nclicks`

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

# Summarise the model
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
##
## Residual standard error: 0.8692 on 97 degrees of freedom
## Multiple R-squared:  0.06543,    Adjusted R-squared:  0.04616
## F-statistic: 3.395 on 2 and 97 DF,  p-value: 0.03756
```

10 Model results

- From the output, we can see that
  - $\hat{\beta}_0 = 1.46$ (intercept)
  - $\hat{\beta}_1 = 0.09$ (`lecture` coefficient)
  - $\hat{\beta}_2 = 0.01$ (`nclicks` coefficient)

11 Plugging the estimates back into the formula

- The result indicates that the following formula can be used to describe how a persons grade is predicted by their lecture attendance and course material download behaviour:

`grade` $= 1.46 + 0.09 \times$ `lecture` $+ 0.01 \times$ `nclicks`

- And, because the regression coefficients of $\hat{\beta}_1$ (`lecture`) and $\hat{\beta}_2$ (`nclicks`) are both positive we can surmise that these predictors have a positive impact on `grade`.
- If you had data on students `nclicks` and `lecture` attendance, you could use this to estimate their grade, based on the multiple regression model.

12 Predicting from new data

- **Warning**: If you want to pass new data to your multiple regression model the variable names have to match exactly. R is unforgiving when it comes to labels, so match sure both the name and text case is the same in your data and the model.

```r
# FYI: A 'tribble' is a way to make a tibble by rows, rather than by columns

new_data <-
  tribble(~lecture, ~nclicks,
          3, 70,
          10, 130,
          0, 20,
          5, 100)
```

- Now that we've created our table `new_data`, we can pass it to mutate and predict() to add a vector with the predictions for Y (`grade`).
- Remember we have already created a model called **`my_model`** based on the composition:

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

```r
# Add predicted grade vector using `predict` function
new_data |>
  mutate(predicted_grade = predict(my_model, new_data))
```

```
## # A tibble: 4 x 3
##   lecture nclicks predicted_grade
##     <dbl>   <dbl>           <dbl>
## 1       3      70            2.09
## 2      10     130            3.03
## 3       0      20            1.56
## 4       5     100            2.42
```

13 Visualising Partial effects

- Each regression coefficient parameter estimate indicates the *partial effect* of that variable; i.e., that variable's effect holding all other variables constant.
- You can visualise partial effects using `predict` by
  - making a table with varying values of the focal predictor and filling all other predictors with their mean values (i.e., keep them constant)

14 Visualising the partial effect of `lecture` on `grade` holding `nclicks` constant

- Remember, `lecture` is an integer from 0:10, so we want to create a vector that includes each of these levels.
- To keep `nclicks` constant, let's create a vector that only contains the mean value for `nclicks`.

14.1 R code for partial effects

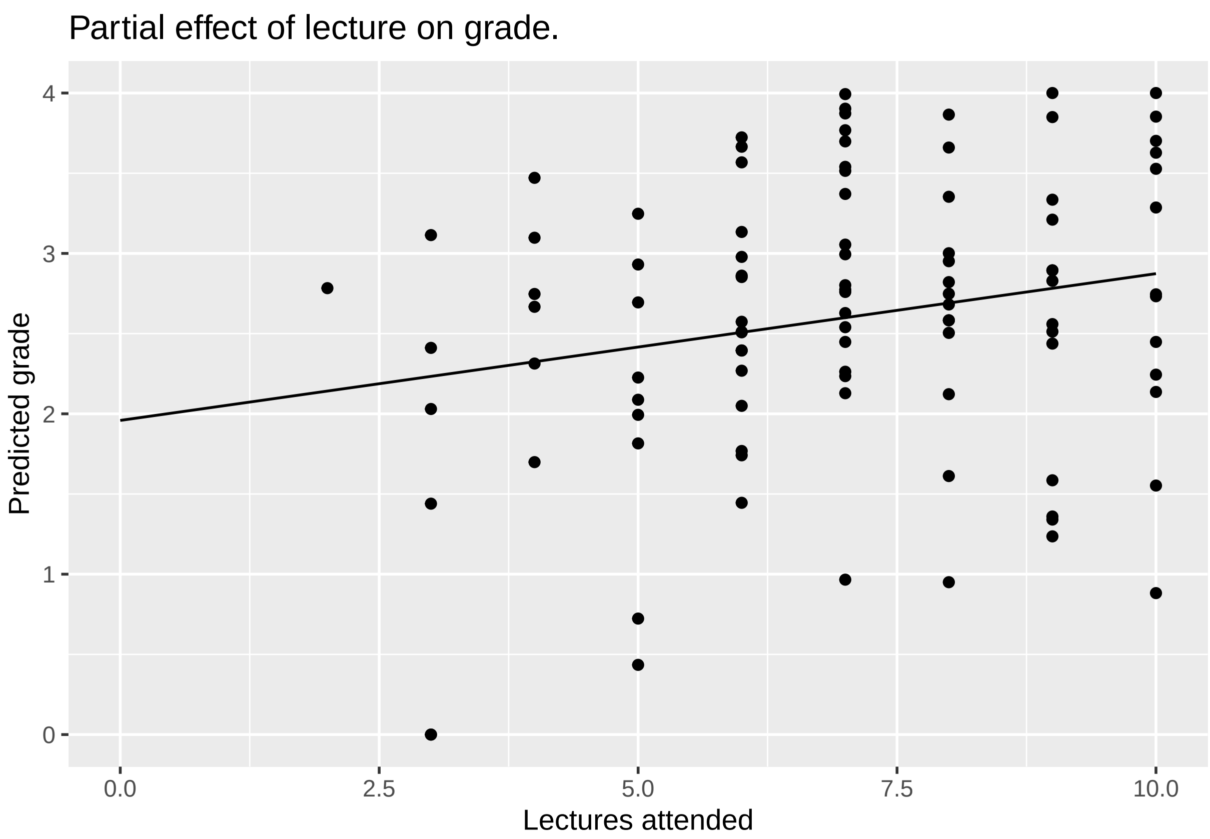
```
# Create vector containing nclicks mean
nclicks_mean <-
  grades |>          # take the grades dataset
  pull(nclicks) |>  # extract single column from df as a vector
  mean()

# Create new data for prediction
new_lecture <-
  tibble(lecture = 0:10,          # create vector containing each level of lecture
         nclicks = nclicks_mean) # add vector of nclicks mean

# Add predicted grades vector controlling for effects of nclicks
new_lecture2 <-
  new_lecture |>
  mutate(grade = predict(my_model, new_lecture))

# Present data
new_lecture2
```

```
## # A tibble: 11 x 3
##    lecture nclicks grade
##      <int>   <dbl> <dbl>
##  1       0    98.3  1.96
##  2       1    98.3  2.05
##  3       2    98.3  2.14
##  4       3    98.3  2.23
##  5       4    98.3  2.32
##  6       5    98.3  2.42
##  7       6    98.3  2.51
```

```
## 8       7    98.3  2.60
## 9       8    98.3  2.69
## 10      9    98.3  2.78
## 11     10    98.3  2.87
```

15 Plot Partial effets

```r
# Plot partial effect of lecture on grade
# Holding `nclicks` constant
ggplot(grades, aes(lecture, grade)) +
  geom_point() +
  geom_line(data = new_lecture2) + # add your
  labs(title = "Partial effect of lecture on grade.",
       x = "Lectures attended",
       y = "Predicted grade")
```



15.1 A word on partial effects plots

- Partial effects plots are meaningful when there are no interactions in the model between the focal predictor and any other predictors.

- This is because, when there are interactions, the partial effect of a focal predictor X_i will differ across the values of other predictors it interacts with.

16 Standardising Coefficients

- Part of multiple regression modelling is determining which of the predictors in your model matter the most when predicting Y.
- In the analysis above, all of the $\hat{\beta}$ (coefficient estimates) come from different scales, so comparing their values is meaningless.
- One way you can convert these scales into something comparable is to convert them into **z-scores**.

$$z = \frac{X - \mu_x}{\sigma_x}$$

16.1 Z-scores

- z-scores represent how far a value of X is from the sample mean (μ_x) in standard deviations (σ_x).
- When you re-scale using z-scores the mean of the scale is set to 0.
- So, a z-score of 1 ($z = 1$) means that that particular score for X is one standard deviation higher than the mean, and -1 would indicate a score 1 standard deviation below the mean.
- Z-scores offer a means to compare data that come from different populations by converting the values to a standard normal distribution (a distribution with a mean of 0 and SD = 1).

17 Rescaling predictors

```r
# Create new object with scaled z-score data vectors
grades2 <-
  grades |>
  mutate(lecture_c =
           (lecture - mean(lecture)) / sd(lecture), # apply z-score formula
         nclicks_c =
           (nclicks - mean(nclicks)) / sd(nclicks))

# Examine the data
head(grades2)
```

```
## # A tibble: 6 x 6
##   grade   GPA lecture nclicks lecture_c nclicks_c
```

```
##    <dbl> <dbl>   <int>   <int>    <dbl>    <dbl>
## 1   2.40  1.13       6      88   -0.484   -0.666
## 2   3.67  0.971      6      96   -0.484   -0.150
## 3   2.85  3.34       6     123   -0.484    1.59
## 4   1.36  2.76       9      99    0.982    0.0439
## 5   2.31  1.02       4      66   -1.46    -2.09
## 6   2.58  0.841      8      99    0.493    0.0439
```

- Now let's fit a model using our z-scores for equal comparison

```r
my_model_scaled <-
  lm(grade ~ lecture_c + nclicks_c,
     grades2)

# Summarise the model
summary(my_model_scaled)
```

```
##
## 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 <0.0000000000000002 ***
## 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
##
## Residual standard error: 0.8692 on 97 degrees of freedom
## Multiple R-squared:  0.06543,    Adjusted R-squared:  0.04616
## F-statistic: 3.395 on 2 and 97 DF,  p-value: 0.03756
```

18 Interpretation

- Now that we have scaled the data we can compare the coefficient estimates
- The model output indicates that `lecture_c` actually had more of an impact on `grade`, with each SD increase in lecture_c grade increased by 0.19 (i.e., $\hat{\beta_1} = 0.19$).
- This is compared to our un-scaled model where the estimate was 0.091 (i.e., $\hat{\beta_1} = 0.09$)

19 Model Comparison

- You may also want to check whether a predictor variable significantly affects the dependent (or response) variable, over and above the effect of one of your control variables.
- We saw above that the model including `lecture` and `nclicks` was significant, $F(2, 97) = 3.395, p = 0.038$.

- The null hypothesis for a multiple regression model represents a model where all of the coefficients (other than the intercept) are zero: $H_0 : \beta_1 - \beta_2 = ... = \beta_m = 0$ OR $Y_i = \beta_0$
- Put differently, your best prediction of Y is simply its mean (μ_y), and the X predictor variables have no effect on Y.
- The regression model above rejects H_0, indicating that `lecture` and `nclicks` can be used to predict `grade`.

20 Reconceptualising the question

- It is possible that better students (who are more likely to attend lectures and download online course content) are simply more likely to get better grades.
- If this is true, than the relationship between `lecture`, `nclicks`, and `grade` would be mediated by student quality.
- So, the question becomes; **are `lecture` and `nclicks` associated with better grades above and beyond student ability, indicated by `GPA`**.

21 Running model comparisons

1. Estimate a model containing any control predictors, excluding the focal predictors.
2. Estimate a model containing the control predictors, including the focal predictors.
3. Compare the two models using `anova`

22 R Code for model comparisons

```
# Control model
m1 <-
  lm(grade ~ GPA, grades)

# Focal predictor model
m2 <-
  lm(grade ~ GPA + lecture + nclicks, grades)

# Run the model comparison
anova(m1, m2)
```

```
## 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
```

23 Interpretation of model comparisons

- H_0 states that we can predict `grade` from `GPA`, just as well as we can from `GPA`, `lecture`, and `nclicks`.
- H_0 will be rejected if the inclusion of `lecture` and `nclicks` (i.e., in the focal predictor model) leads to a substantial reduction in the residual sum of squares.
- This would indicate that their inclusion helps to signidicantly reduce the amount of unexplained variance in the model.
- The result $F(2, 96) = 1.308, p = 0.275$ shows that our control variable model is as good at explaining the results as our focal predictor model.
- So, `lecture` and `nclicks` do not predict better grades more so than `GPA` alone.

24 What did we cover today

- Equations/formula for multiple regression
- Worked example using the “grades.csv” dataset from PsyTeachR
- The `predict` function

- Calculating and visualising partial effects
- Comparing standard models and non-standardised models

25 Tutorial exercise for this week

- Visualize the partial effect of `nclicks` on `grade`.

26 Reading

Learning Statistical Models Through Simulation in R: Chapter 4 Multiple Regression