

# Math Final

Matthew Vanaman

05-21-2019

## 7.11

(a)

$$\frac{\partial}{\partial x_1} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4} = \beta_1.$$

(b)

As long as the  $x$  variable is not exponentiated or multiplied, the coefficient is the partial derivative. This is not true for exponentiated terms. Take for example:

$$\begin{aligned} & \frac{d}{dx}(a + \beta x^2), \\ &= \frac{d}{dx}(a) + \frac{d}{dx}(\beta x^2), \\ &= \beta \frac{d}{dx}(x^2), \\ &= \beta(2x^{2-1}), \\ &= 2\beta x. \end{aligned}$$

This derivative does not yield the coefficient by itself, so you cannot interpret the coefficient as a straightforward linear relationship between  $x$  and  $y$  in the case of exponentiated terms. When the slope is exponential, the partial derivative gives you the instantaneous rate of change at each level of  $x$  which, unlike in the case of (a), will be different at each level of  $x$ .

Multiplicative relationships also cannot be interpreted as a straightforward linear relationship. Consider a model with two predictors,  $x_1$  and  $x_2$ , and their interaction:

$$\frac{\partial}{\partial x_1}(a + \beta_1 x_1 + \beta_2 x_2 + \beta_{12}(x_1 \times x_2)) = \beta_1 + \beta_{12} x_2.$$

Here, the first derivative with respect to  $x_1$  is a function of  $x_2$ . If there is an interaction, this partial derivative will change when evaluated across differing  $x_2$  values. The key thing to notice is that because this function incorporates information about  $x_2$ , it does not describe an exclusive relationship between  $x_1$  and  $y$  like function (a) does. Therefore, when an interaction term is present, the first partial derivative does not yield the regression coefficient  $\beta_1$ . However, if this derivative does not change across differing levels of  $x_2$ , there is no interaction. In cases where there is no interaction (i.e. in cases where  $\beta_{12} x_2$  comes out to zero), this function *would* describe the linear relationship between  $x_1$  and  $y$  in the way function (a) does. This would also be true whenever  $x_2 = 0$ . However, mathematically, this derivative does not directly represent the coefficient one sees in regression output for  $x_1$  in a main-effects only model, nor does it yield the coefficient for the interaction.

To get the coefficient for the interaction, you need to take the *cross-partial* derivative with respect to both  $x_1$  and  $x_2$ :

$$\frac{\partial^2}{\partial x_1 \partial x_2}(a + \beta_1 x_1 + \beta_2 x_2 + \beta_{12}(x_1 \times x_2)) = \beta_{12}.$$

Importantly, this coefficient does not describe an exclusive relationship between  $x_1$  and  $y$  since it is a function of both  $x_1$  and  $x_2$ . The cross-partial derivative introduces a third slope, which is the slope between  $x_1$  and  $y$  spanning across all levels of  $x_2$ . If this slope is zero, you have no interaction.

In sum, the coefficient seen in regression output describes a linear relationship between  $x$  and  $y$  in a linear main-effects model but the meaning of this coefficient changes when  $x$  is exponentiated or multiplied by another independent variable.

(c)

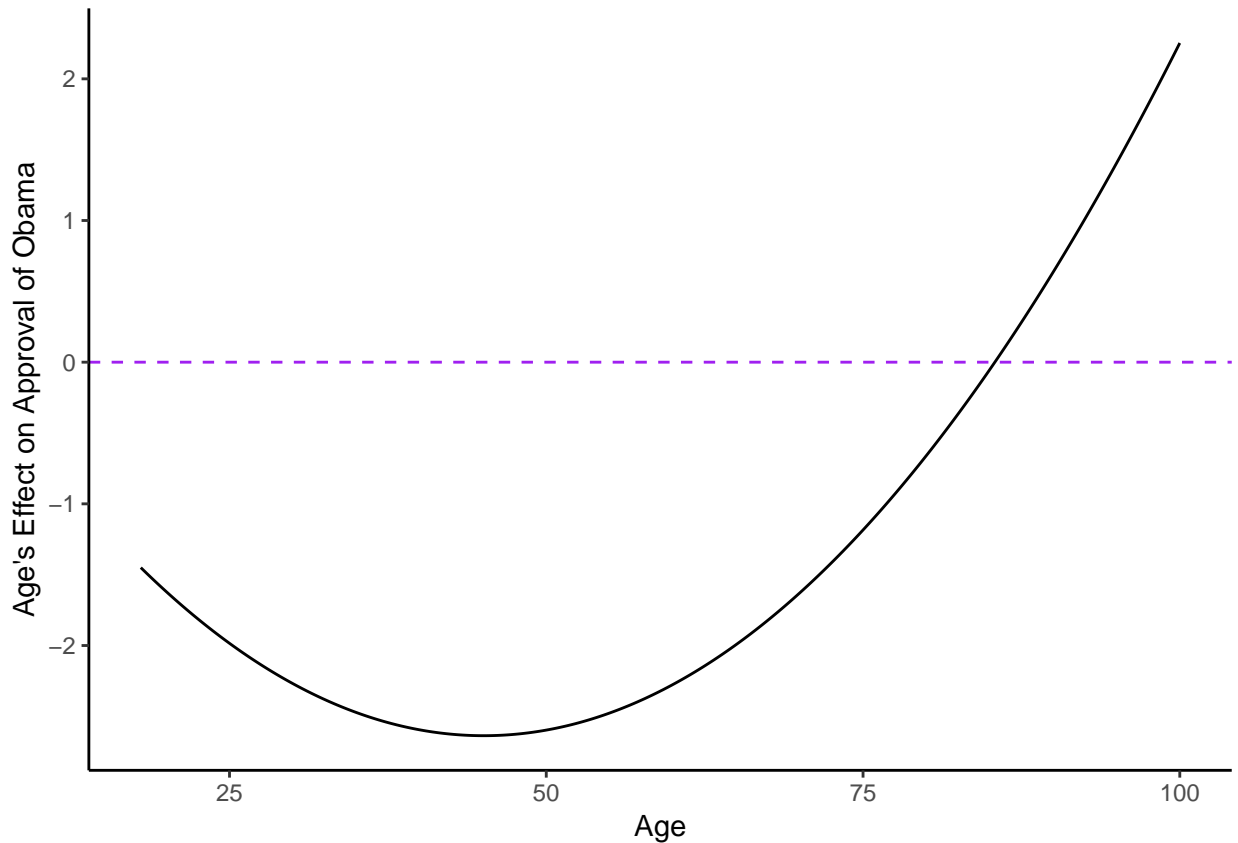
$$\frac{\partial}{\partial x_2} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4} = \beta_2 + 2\beta_3 x_{i2} + 3\beta_4 x_{i2}^2.$$

(d)

On the x-axis, you will age. On the y-axis, you will have:

$$\begin{aligned} &0.653 + 2(0.073)x_{i2} + 3(0.00054)x_{i2}^2, \\ &= 0.6530.146x_{i2} + 0.00162x_{i2}^2. \end{aligned}$$

Voting age starts at 18, so make age run from 18 to 100, let's say.



There is a curvilinear relationship such that generally speaking, age is negatively associated with approval. That is, the slope between age and approval is negative everywhere except for those older than around 80 years old or so. At that age, the effect turns around, and age becomes positively predictive of approval. The effect of age on approval is most negative at middle age.

(e)

$$\frac{\partial}{\partial x_4} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4} = \beta_6 + \beta_7 x_{i3}.$$

(f)

$$= \beta_6 + \beta_7 x_{i3} = 1 + 3x_{i3}.$$

Dummy code sex and make male = 0 and female = 1:

$$\text{male} = 1 + 3(0) = 1.$$

$$\text{female} = 1 + 3(1) = 3.$$

(g)

In both males and females, the derivative is positive in sign, indicating that there is a positive slope between pro-choice views and approval of Obama. However, there is also a sex difference: for females, the effect is 3 whereas for males it is 1; what this means is that the slope between abortion views and approval of Obama is steeper for women than for men. Because there is a difference in slopes of sex, we can say there is an interaction. In other words, there is a compounding effect of sex such that when you move from the male to female level, you see an increase in the strength of the relationship between abortion views and approval of Obama (descriptively speaking anyway).

## 9.10

(a)

$$X'Y = \begin{bmatrix} 68 \\ 121 \\ 179 \end{bmatrix}.$$

(b)

$$X'X = \begin{bmatrix} 10 & 25 & 22 \\ 25 & 165 & 71 \\ 22 & 71 & 178 \end{bmatrix}.$$

(c)

$$(X'X)^{-1} = \begin{bmatrix} 0.187 & -0.022 & -0.014 \\ -0.022 & 0.01 & -0.001 \\ -0.014 & -0.001 & 0.008 \end{bmatrix}$$

(d)

$$\hat{\beta} = \begin{bmatrix} 7.468 \\ -0.524 \\ 0.291 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

## Check with Regression

Table 1

Predictor	<i>b</i>	95% CI	<i>t</i> (7)	<i>p</i>
Intercept	7.47	[4.75, 10.18]	6.51	< .001
Sanctions	-0.52	[-1.15, 0.10]	-1.98	.089
Riots	0.29	[-0.27, 0.85]	1.24	.256

## 10.9

### (a)

#### Characterization:

Social capital describes the level of cohesion members of a community have. It can be characterized as falling along two dimensions: trust in other members of the community and willingness to cooperate with others. That is, social capital is a function of both ambient levels of trust in a community as well as the general tendency to cooperate civically, such as by returning found money or being willing to contribute to public welfare through taxes and fares. Communities with the most social capital will on average score highest on both dimensions while those with lowest social capital will score lowest.

#### Representation:

To represent this construct, we'd need to identify indicators of the construct. What kinds of thoughts, experiences, and behaviors do individuals of a community with high social capital tend to do? The items should reflect these. Trust is represented by a single-item measure where the top of the scale represents the attitude that most people can be trusted and the low end represents the attitude that most people cannot be trusted. Those high in the trust dimension of social capital should score toward the high end of the scale and those low in trust should score toward the low end of the scale. Civic cooperation is represented by a set of statements describing situations where one could act for the benefit of the community but chose not to. The high end of the scale represents the attitude that these behaviors are never acceptable while the low represents the attitude that they are always acceptable. Those high in civic cooperation should score toward the high end of the scale and those low should score on the low end.

#### Measurement:

The measurement model is a principal components analysis in this case.

### (b)

Eigenvalues: 2.41, 11.42.

Eigenvectors:  $\begin{bmatrix} 0.75 \\ -0.65 \end{bmatrix}, \begin{bmatrix} 0.64 \\ 0.73 \end{bmatrix}$ .

### (c)

Because the eigenvalues are on the variance metric, the variance explained is a straightforward proportion:

$$\frac{11.42}{11.42 + 2.41} = 0.83\%.$$

(d)

If we go with the single-component solution, then social capital is:

$$0.64\text{Trust} + 0.752\text{Civic Cooperation}.$$

I'm not sure how to tell which values of the eigenvector go with which dimension, but it probably doesn't matter since we went with a unidimensional solution?

Anyway, if you want to index the individuals in the dataset, you would calculate their scores with the formula above:

Table 2

Person	Trust	Civic Cooperation	Formula	Score
1	4	1	$0.64(4) + 0.752(1)$	4.06
2	7	10	$0.64(7) + 0.752(10)$	12
3	3	6	$0.64(3) + 0.752(6)$	6.43
4	8	9	$0.64(8) + 0.752(9)$	11.89
5	9	7	$0.64(9) + 0.752(7)$	11.02
6	2	3	$0.64(2) + 0.752(3)$	3.54
7	9	6	$0.64(9) + 0.752(6)$	10.27
8	5	3	$0.64(5) + 0.752(3)$	5.46
9	6	4	$0.64(6) + 0.752(4)$	6.85
10	8	8	$0.64(8) + 0.752(8)$	11.14



## Appendix

### 7.11

(a)

**Work:**

$$\frac{\partial}{\partial x_1} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4},$$

Derivative of a constant is zero, so drop  $\alpha$ :

$$= \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4},$$

Break up across addition and subtraction, then bring out constants:

$$= \beta_1 \frac{\partial}{\partial x_1} x_{i1} + \beta_2 \frac{\partial}{\partial x_1} x_{i2} + \beta_3 \frac{\partial}{\partial x_1} x_{i2}^2 + \beta_4 \frac{\partial}{\partial x_1} x_{i2}^3 + \beta_5 \frac{\partial}{\partial x_1} x_{i3} + \beta_6 \frac{\partial}{\partial x_1} x_{i4} + \beta_7 \frac{\partial}{\partial x_1} x_{i3} \frac{\partial}{\partial x_1} x_{i4},$$

All other independent variables are treated as constants:

$$= \frac{\partial}{\partial x_1} \beta_1 x_{i1} + 1(0) + 1(0) + 1(0) + 1(0) + 1(0) + 1(0)(0),$$

Bring the constant out:

$$= \beta_1 \frac{\partial}{\partial x_1} x_{i1} = \beta_1(1) = \beta_1.$$

(b)

NA

(c)

**Work:**

$$\frac{\partial}{\partial x_2} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4},$$

Derivative of a constant is zero, so drop  $\alpha$ :

$$= \frac{\partial}{\partial x_2} (\beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4}),$$

Break up across addition and subtraction, then again across multiplication:

$$= \frac{\partial}{\partial x_2} \beta_1 x_{i1} + \frac{\partial}{\partial x_2} \beta_2 x_{i2} + \frac{\partial}{\partial x_2} \beta_3 x_{i2}^2 + \frac{\partial}{\partial x_2} \beta_4 x_{i2}^3 + \frac{\partial}{\partial x_2} \beta_5 x_{i3} + \frac{\partial}{\partial x_2} \beta_6 x_{i4} + \frac{\partial}{\partial x_2} \beta_7 x_{i3} x_{i4},$$

The independent variables all get treated as constants:

$$= 0 + \frac{\partial}{\partial x_2} \beta_2 x_{i2} + \frac{\partial}{\partial x_2} \beta_3 x_{i2}^2 + \frac{\partial}{\partial x_2} \beta_4 x_{i2}^3 + 0 + 0 + 0 + 0,$$

$$= \beta_2 \frac{\partial}{\partial x_2} x_{i2} + \beta_3 \frac{\partial}{\partial x_2} x_{i2}^2 + \beta_4 \frac{\partial}{\partial x_2} x_{i2}^3,$$

$$= \beta_2 + 2\beta_3 x_{i2} + 3\beta_4 x_{i2}^2.$$

(d)

```
x <- seq(18, 100, length=10000)
pd <- 0.653 - 0.146*x + 0.00162*x^2
df <- as.data.frame(cbind(x,pd))
ggplot2::ggplot(df, aes(x)) +
  geom_line(aes(y = pd)) +
  ylab("Age's Effect on Approval of Obama") + xlab("Age") +
  labs(color = "Function") +
  geom_hline(yintercept=0, linetype="dashed", color = "purple") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"))
```

(e)

Work:

$$\frac{\partial}{\partial x_4} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4},$$

Derivative of a constant is zero, so drop  $\alpha$ :

$$\begin{aligned} &= \frac{\partial}{\partial x_4} (\beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i2}^2 + \beta_4 x_{i2}^3 + \beta_5 x_{i3} + \beta_6 x_{i4} + \beta_7 x_{i3} x_{i4}), \\ &= \frac{\partial}{\partial x_4} (\beta_1 x_{i1}) + \frac{\partial}{\partial x_4} \beta_2 x_{i2} + \frac{\partial}{\partial x_4} \beta_3 x_{i2}^2 + \frac{\partial}{\partial x_4} \beta_4 x_{i2}^3 + \frac{\partial}{\partial x_4} \beta_5 x_{i3} + \frac{\partial}{\partial x_4} \beta_6 x_{i4} + \frac{\partial}{\partial x_4} \beta_7 x_{i3} x_{i4}, \\ &= 0 + 0 + 0 + 0 + 0 + 0 + \frac{\partial}{\partial x_4} \beta_6 x_{i4} + \frac{\partial}{\partial x_4} \beta_7 x_{i3} x_{i4}, \\ &= \beta_6 \frac{\partial}{\partial x_4} x_{i4} + \beta_7 x_{i3} \frac{\partial}{\partial x_4} x_{i4}, \\ &= \beta_6 + \beta_7 x_{i3}. \end{aligned}$$

(f)

NA

(g)

NA

## 9.10

```
print.matrix <- function(m) {
  write.table(format(m, justify="right"),
              row.names=F, col.names=F, quote=F)
}
# This function lets you call R matrices while in latex math mode
matex <- function(x) {
  begin <- "\\begin{bmatrix}"
  end <- "\\end{bmatrix}"
  X <-
  apply(x, 1, function(x) {
```



```

    paste(
      paste(x, collapse = "&"),
      "\\\\"
    )
  })
  paste(c(begin, X, end), collapse = "")
}

y <- matrix(c(10, 7, 2, 8, 10, 6, 2, 10, 8, 5),
            nrow=10, ncol=1)
x <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
              2, 1, 2, 0, 1, 3, 11, 0, 5, 0,
              5, 1, 0, 2, 0, 0, 0, 0, 12, 2),
            ncol = 3, nrow = 10)
x_prime <- t(x)
B <- matrix(c("\\alpha", "\\beta_1", "\\beta_2"), nrow=3, ncol=1)
E <- matrix(c("\\varepsilon_{\\textrm{USA}}",
              "\\varepsilon_{\\textrm{Peru}}",
              "\\varepsilon_{\\textrm{Brazil}}",
              "\\varepsilon_{\\textrm{Spain}}",
              "\\varepsilon_{\\textrm{Norway}}",
              "\\varepsilon_{\\textrm{Zimbabwe}}",
              "\\varepsilon_{\\textrm{Turkey}}",
              "\\varepsilon_{\\textrm{Japan}}",
              "\\varepsilon_{\\textrm{India}}",
              "\\varepsilon_{\\textrm{Sri \\ Lanka}}"),
            nrow=10, ncol=1)

```

(a)

Work:

$$\begin{aligned}
 X'Y &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 2 & 0 & 1 & 3 & 11 & 0 & 5 & 0 \\ 5 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 12 & 2 \end{bmatrix} \begin{bmatrix} 10 \\ 7 \\ 2 \\ 8 \\ 10 \\ 6 \\ 2 \\ 10 \\ 8 \\ 5 \end{bmatrix}, \\
 &= 1(10) + 1(7) + 1(2) + 1(8) + 1(10) + 1(6) + 1(2) + 1(10) + 1(8) + 1(5) = 68, \\
 &\quad \text{etc...} \\
 &= \begin{bmatrix} 68 \\ 121 \\ 179 \end{bmatrix}.
 \end{aligned}$$

(b)

Work:

$$X'X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 2 & 0 & 1 & 3 & 11 & 0 & 5 & 0 \\ 5 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 12 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 5 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 3 & 0 \\ 1 & 11 & 0 \\ 1 & 0 & 0 \\ 1 & 5 & 12 \\ 1 & 0 & 2 \end{bmatrix},$$

$$1(1) + 1(1) + 1(1) + 1(1) + 1(1) + 1(1) + 1(1) + 1(1) + 1(1) + 1(1) = 10,$$

etc...

$$\begin{bmatrix} 10 & 25 & 22 \\ 25 & 165 & 71 \\ 22 & 71 & 178 \end{bmatrix}.$$

```
# check work
x_prime_times_x <- x_prime %*% x
```

(c)

```
# 2 x 2's for computing minor elements
e_11 <- x_prime_times_x[-1,-1] # rm r1 & c1
e_12 <- x_prime_times_x[-1,-2] # rm r1 & c2
e_13 <- x_prime_times_x[-1,-3] # rm r1 & c3
e_21 <- x_prime_times_x[-2,-1] # rm r2 & c1
e_22 <- x_prime_times_x[-2,-2] # rm r2 & c2
e_23 <- x_prime_times_x[-2,-3] # rm r2 & c3
e_31 <- x_prime_times_x[-3,-1] # rm r3 & c1
e_32 <- x_prime_times_x[-3,-2] # rm r3 & c2
e_33 <- x_prime_times_x[-3,-3] # rm r3 & c3

MEs <- matrix(c(det(e_11), det(e_12), det(e_13),
                det(e_21), det(e_22), det(e_23),
                det(e_31), det(e_32), det(e_33)),
              nrow = 3, ncol = 3)

# make cofactor matrix
cofactor <- matrix(c(det(e_11), (-1 * det(e_12)), det(e_13),
                    (-1 * det(e_21)), det(e_22), (-1 * det(e_23)),
                    det(e_31), (-1 * det(e_32)), det(e_33)),
                  nrow = 3, ncol = 3)

xpx_col_1 <- matrix(x_prime_times_x[,2])
cof_col1 <- matrix(cofactor[,2])
prod <- xpx_col_1 * cof_col1
sum <- sum(prod)
inverse <- (1/sum) * cofactor
```

$$(X'X)^{-1} = \text{rmatrix}(\text{round}(\text{inverse}, 3))$$

Work:

```
# modify the function to print determinants
print.matrix <- function(m) {
  write.table(format(m, justify="right"),
              row.names=F, col.names=F, quote=F)
}
# This function lets you call R matrices while in latex math mode
matex_det <- function(x) {
  begin <- "\\begin{vmatrix}"
  end <- "\\end{vmatrix}"
  X <-
    apply(x, 1, function(x) {
      paste(
        paste(x, collapse = "&"),
        "\\\\"
      )
    })
  paste(c(begin, X, end), collapse = "")
}

# 2 x 2's for computing minor elements
e_11 <- x_prime_times_x[-1,-1] # rm r1 & c1
e_12 <- x_prime_times_x[-1,-2] # rm r1 & c2
e_13 <- x_prime_times_x[-1,-3] # rm r1 & c3
e_21 <- x_prime_times_x[-2,-1] # rm r2 & c1
e_22 <- x_prime_times_x[-2,-2] # rm r2 & c2
e_23 <- x_prime_times_x[-2,-3] # rm r2 & c3
e_31 <- x_prime_times_x[-3,-1] # rm r3 & c1
e_32 <- x_prime_times_x[-3,-2] # rm r3 & c2
e_33 <- x_prime_times_x[-3,-3] # rm r3 & c3

# check work
inverse_x_xprime <- round(solve(x_prime %*% x), 3)
```

Follow the formula for the inverse of a large square matrix:

$$\frac{1}{|A|} \text{adj}(A) = \frac{1}{|X'X|} \text{adj}(X'X).$$

First we have to get the adjoint matrix. To get the adjoint matrix, we have to find the minor elements, followed by the cofactor matrix, then we can derive the adjoint matrix.

To get the minor elements, we find the determinant of each possible 2 x 2 matrix. The determinant matrix is:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc,$$

which we will have to do for all of the possible matrices. Starting with element (1,1):

$$e_{11} = \begin{vmatrix} 165 & 71 \\ 71 & 178 \end{vmatrix} = 165(178) - 71(71) = 24,329.$$

We continue this all the way down the line:

$$e_{12} = \begin{vmatrix} 25 & 71 \\ 22 & 178 \end{vmatrix} = 2888.$$

$$e_{13} = \begin{vmatrix} 25 & 22 \\ 165 & 71 \end{vmatrix} = -1855.$$

$$e_{21} = \begin{vmatrix} 25 & 22 \\ 71 & 178 \end{vmatrix} = 2888.$$

$$e_{22} = \begin{vmatrix} 10 & 22 \\ 22 & 178 \end{vmatrix} = 1296.$$

$$e_{23} = \begin{vmatrix} 10 & 25 \\ 22 & 71 \end{vmatrix} = 160.$$

$$e_{31} = \begin{vmatrix} 25 & 22 \\ 165 & 71 \end{vmatrix} = -1855.$$

$$e_{32} = \begin{vmatrix} 10 & 25 \\ 22 & 71 \end{vmatrix} = 160.$$

$$e_{33} = \begin{vmatrix} 10 & 25 \\ 25 & 165 \end{vmatrix} = 1025.$$

```
# make matrix of minor elements
MEs <- matrix(c(det(e_11), det(e_12), det(e_13),
                det(e_21), det(e_22), det(e_23),
                det(e_31), det(e_32), det(e_33)),
              nrow = 3, ncol = 3)

# make cofactor matrix
cofactor <- matrix(c(det(e_11), (-1 * det(e_12)), det(e_13),
                    (-1 * det(e_21)), det(e_22), (-1 * det(e_23)),
                    det(e_31), (-1 * det(e_32)), det(e_33)),
                  nrow = 3, ncol = 3)
```

$$\text{Minor Elements Matrix} = \begin{bmatrix} 24329 & 2888 & -1855 \\ 2888 & 1296 & 160 \\ -1855 & 160 & 1025 \end{bmatrix}.$$

To get the cofactor matrix, you just multiple the cells by -1 if their row and columns number add up to an odd number:

$$\text{Cofactor Matrix} = \begin{bmatrix} 24329 & -2888 & -1855 \\ -2888 & 1296 & -160 \\ -1855 & -160 & 1025 \end{bmatrix}.$$

Lastly, you derive the adjoint matrix by transposing the cofactor matrix. But since it's symmetric, the cofactor matrix equals the adjoint matrix:

$$\text{Adjoint Matrix} = \begin{bmatrix} 24329 & -2888 & -1855 \\ -2888 & 1296 & -160 \\ -1855 & -160 & 1025 \end{bmatrix}.$$

Now that we have the adjoint matrix, we use it to find the determinant of the  $X'X$  matrix. We do this by multiplying a row or column of  $X'X$  by its corresponding elements in the cofactor matrix and then sum the product.

```
# check work
xpx_col_1 <- matrix(x_prime_times_x[,2])
cof_col1 <- matrix(cofactor[,2])
prod <- xpx_col_1 * cof_col1
sum <- sum(prod)
```

$$\begin{bmatrix} 25 \\ 165 \\ 71 \end{bmatrix} \times \begin{bmatrix} -2888 \\ 1296 \\ -160 \end{bmatrix} = \sum \begin{bmatrix} -72200 \\ 213840 \\ -11360 \end{bmatrix} = 130280.$$

Last, now that we have  $|X'X|$ , we can use the formula for matrix inversion:

$$(X'X)^{-1} = \frac{1}{|X'X|} \text{adj}(X'X),$$

$$= \frac{1}{130280} \times \begin{bmatrix} 24329 & -2888 & -1855 \\ -2888 & 1296 & -160 \\ -1855 & -160 & 1025 \end{bmatrix} = \begin{bmatrix} 0.187 & -0.022 & -0.014 \\ -0.022 & 0.01 & -0.001 \\ -0.014 & -0.001 & 0.008 \end{bmatrix}.$$

```
# check work with the inverse function
solve <- solve(x_prime_times_x)
```

(d)

Work:

$$\hat{\beta} = (X'X)^{-1}(X'Y),$$

$$= \begin{bmatrix} 0.187 & -0.022 & -0.014 \\ -0.022 & 0.01 & -0.001 \\ -0.014 & -0.001 & 0.008 \end{bmatrix} \times \begin{bmatrix} 68 \\ 121 \\ 179 \end{bmatrix},$$

$$= \begin{bmatrix} 7.468 \\ -0.524 \\ 0.291 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \end{bmatrix}.$$

Check with Regression

```
demo_data <- data.frame(c(c("USA", "Peru", "Brazil", "Spain",
                           "Norway", "Zimbabwe", "Turkey",
                           "Japan", "India", "Sri Lanka")),
                        y, x[,2], x[,3])

colnames(demo_data) <- c("Country", "Democracy",
                        "Sanctions", "Riots")

fit <- summary(lm(Democracy ~ Sanctions + Riots,
                  data = demo_data))

tab <- papaja::apa_print.summary.lm(fit)
papaja::apa_table(tab$table, placement = "h")
```

## 10.9

(a)

NA

(b)

Eigenvalues: 2.41, 11.42.

Eigenvectors:  $\begin{bmatrix} 0.75 \\ -0.65 \end{bmatrix}, \begin{bmatrix} 0.64 \\ 0.73 \end{bmatrix}$ .

**Work:**

The covariance matrix of the data is:

$$\begin{bmatrix} 6.32 & 4.47 \\ 4.47 & 7.51 \end{bmatrix}$$

Find the eigenvalues and unit eigenvectors of the covariance matrix.

There are 4 steps for finding the eigenvalues of a square matrix. The first step is to write out the matrix  $(A - \lambda I)$ , then find the determinant of this matrix  $|A - \lambda I|$  in terms of  $\lambda$ :

$$\begin{aligned} \begin{bmatrix} 6.32 & 4.47 \\ 4.47 & 7.51 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \\ \left( \begin{bmatrix} 6.32 & 4.47 \\ 4.47 & 7.51 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \left( \begin{bmatrix} 6.32 & 4.47 \\ 4.47 & 7.51 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \left( \begin{bmatrix} 6.32 - \lambda & 4.47 \\ 4.47 & 7.51 - \lambda \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \begin{bmatrix} 6.32 - \lambda & 4.47 \\ 4.47 & 7.51 - \lambda \end{bmatrix} &. \end{aligned}$$

Set the determinant equal to 0, then solve for  $\lambda$ :

$$\begin{aligned} \begin{vmatrix} 6.32 - \lambda & 4.47 \\ 4.47 & 7.51 - \lambda \end{vmatrix} &= 0, \\ (6.32\lambda)(7.51\lambda)4.47^2 &= 0, \\ (6.32\lambda)(7.51\lambda)4.47^2 &= 0, \end{aligned}$$

Let python do the algebra - eigenvalues equal:

```
# Check Work
import sympy
from sympy import *
from sympy import Symbol
x = Symbol('x')
#Solve
lamb = solveset(Eq((6.32 - x)*(7.51 - x) - 4.47**2, 0), x)
for a in preorder_traversal(lamb):
    if isinstance(a, Float):
        lamb = lamb.subs(a, round(a, 2))
print("solveset(Eq((6.32 - x)*(7.51 - x) - 4.47**2, 0), x)")

print(lamb)
```

To get the eigenvectors, start by substituting these back in for  $\lambda$  and reduce them:

$$\begin{aligned} & \begin{bmatrix} 6.32 - \lambda & 4.47 \\ 4.47 & 7.51 - \lambda \end{bmatrix}, \\ &= \begin{bmatrix} 6.32 - 2.41 & 4.47 \\ 4.47 & 7.51 - 2.41 \end{bmatrix}, \\ &= \begin{bmatrix} 3.91 & 4.47 \\ 4.47 & 5.10 \end{bmatrix}. \end{aligned}$$

Multiply the second row by -0.87 ( $\frac{-4.47}{5.10} = -0.87$ ):

$$= \begin{bmatrix} 0 & 0 \\ 4.47 & 5.10 \end{bmatrix}.$$

Because we made  $(A - \lambda I) = 0$  earlier, we know the system of equations will be:

$$\begin{aligned} & \begin{bmatrix} 0 & 0 \\ 4.47 & 5.10 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \\ &= \begin{cases} x = x, \\ 4.47x + 5.10y = 0, \end{cases} \\ &= \begin{cases} x = x, \\ 5.10y = -4.47x, \end{cases} \\ &= \begin{cases} x = x, \\ y = -0.87x, \end{cases} \\ &= \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ -0.87x \end{bmatrix} = x \begin{bmatrix} 1 \\ -0.87 \end{bmatrix}. \end{aligned}$$

To get unit eigenvector, get the vector length and divide the vector by its length:

$$\left| \begin{bmatrix} 1 \\ -0.87 \end{bmatrix} \right| = \sqrt{1^2 + (-0.87)^2} = 1.33.$$

$$\frac{1}{1.33} \begin{bmatrix} 1 \\ -0.87 \end{bmatrix} = \begin{bmatrix} 0.75 \\ -0.65 \end{bmatrix}.$$

Now for the other one:

$$\begin{aligned} &= \begin{bmatrix} 6.32 - 11.42 & 4.47 \\ 4.47 & 7.51 - 11.42 \end{bmatrix}, \\ &= \begin{bmatrix} -5.10 & 4.47 \\ 4.47 & -3.91 \end{bmatrix}, \\ &= \begin{bmatrix} 0 & 0 \\ 4.47 & -3.91 \end{bmatrix}, \\ &= \begin{cases} x = x, \\ 4.47x - 3.91y = 0, \end{cases} \\ &= \begin{cases} x = x, \\ -3.91y = -4.47x, \end{cases} \end{aligned}$$

$$= \begin{cases} x = x, \\ y = 1.14x, \end{cases}$$

$$= \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ 1.14x \end{bmatrix} = x \begin{bmatrix} 1 \\ 1.14 \end{bmatrix}.$$

To get unit eigenvector, get the vector length and divide the vector by its length:

$$\left| \begin{bmatrix} 1 \\ 1.14 \end{bmatrix} \right| = \sqrt{1^2 + 1.14^2} = 1.52.$$

$$\frac{1}{1.56} \begin{bmatrix} 1 \\ 1.14 \end{bmatrix} = \begin{bmatrix} 0.64 \\ 0.73 \end{bmatrix}.$$

```
# check work
ev1 <- ( 1/1.33) * matrix(c(1, -0.87), nrow=2, ncol=1)
ev1 <- round(ev1, 2)
ev2 <- ( 1/1.56) * matrix(c(1, 1.14), nrow=2, ncol=1)
ev2 <- round(ev2, 2)
covar <- matrix(c(6.32, 4.47, 4.47, 7.51),
                 ncol = 2, nrow = 2)
```

(c)

NA

(d)

```
formulas <- c("0.64(4) + 0.752(1)",
              "0.64(7) + 0.752(10)",
              "0.64(3) + 0.752(6)",
              "0.64(8) + 0.752(9)",
              "0.64(9) + 0.752(7)",
              "0.64(2) + 0.752(3)",
              "0.64(9) + 0.752(6)",
              "0.64(5) + 0.752(3)",
              "0.64(6) + 0.752(4)",
              "0.64(8) + 0.752(8)"
             )

scores <- round(c( 0.64 * 4 + 0.752 * 2,
                  0.64 * 7 + 0.752 * 10,
                  0.64 * 3 + 0.752 * 6,
                  0.64 * 8 + 0.752 * 9,
                  0.64 * 9 + 0.752 * 7,
                  0.64 * 2 + 0.752 * 3,
                  0.64 * 9 + 0.752 * 6,
                  0.64 * 5 + 0.752 * 3,
                  0.64 * 6 + 0.752 * 4,
                  0.64 * 8 + 0.752 * 8
                 ), 2)

social.capital <- as.data.frame(matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                                         4, 7, 3, 8, 9, 2, 9, 5, 6, 8,
```



```

      1, 10, 6, 9, 7, 3, 6, 3, 4, 8,
      formulas,
      scores
    ), nrow = 10, ncol = 5)
  )

colnames(social.capital) <- c("Person", "Trust", "Civic Cooperation", "Formula", "Score")
papaja::apa_table(social.capital, placement = "h")

```