BIOS7345 Lab 7

Factor coding schemes

Sarah Lotspeich

2 November 2018

#Introduction We are interested in comparing the `grades` of students in 3 different mathematics `classes`. A sample of 30 students is obtained.

```
n <- 30  #number of students
grades <- rnorm(n, 75, 5)
classes <- sample(c("Class A", "Class B", "Class C"), n, replace = TRUE) %>%
    factor(levels = c("Class A", "Class B", "Class C"))
```

Begin by finding the overall mean grade across all classes

```
mean(grades)
```

```
## [1] 74.24391
```

as well as the mean grade in each class

```
aggregate(grades ~ classes, FUN = mean)
```

```
##   classes   grades
## 1 Class A 73.90756
## 2 Class B 75.10895
## 3 Class C 74.04761
```

Before we begin, a helpful lil page on factor coding schemes.

#Model 1: Reference cell coding Reference cell coding is what we commonly refer to as "dummy variable coding". Since we have $k = 3$ levels of the `classes` variable, how many indicator variables will we need?

There are three levels to the unordered factor variable `classes`

```
table(classes)
```

```
## classes
## Class A Class B Class C
##      11       7      12
```

so we will have 3 - 1 = 2 contrasts.

##Fit the model

```
options(contrasts = c("contr.treatment", "contr.poly"))
mod <- lm(grades ~ classes)
summary(mod)
```

```
##
## Call:
## lm(formula = grades ~ classes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8958 -2.7384 -0.2568  2.3932 10.9884
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      73.908      1.362  54.268   <2e-16 ***
## classesClass B    1.201      2.184   0.550    0.587
## classesClass C    0.140      1.885   0.074    0.941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.517 on 27 degrees of freedom
## Multiple R-squared:  0.01245,    Adjusted R-squared:  -0.0607
## F-statistic: 0.1702 on 2 and 27 DF,  p-value: 0.8444
```

What does your $\boldsymbol{\beta}$ vector look like for this model?

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

##Design matrix Obtain the design matrix for this coding scheme (ignoring replicates) from your `mod` object.

```
# save the unique rows of your design matrix
X = unique(model.matrix(mod))
X
```

```
##   (Intercept) classesClass B classesClass C
## 1           1              0              1
## 2           1              1              0
## 5           1              0              0
```

Stop and think: which group is your reference group?

> The reference group is identified as that for which all indicator variables are equal to 0. In our case, if we assume that the first row is Class A then Class A is our reference group.

##Group means We can obtain the class-specific means from the model parameter estimates by multiplying the unique rows of the design matrix (`X`) by the coefficients.

```
X %*% mod$coefficients
```

```
##       [,1]
## 1 74.04761
## 2 75.10895
## 5 73.90756
```

How do these means compare to those we calculated initially using `aggregate()`?

> The values are the same, but the design matrix outputted by the model has the reference group in the last row (whereas we had it in the first).

So what does this mean? When using reference cell coding, we have

1. $\bar{X}_A = \hat{\beta}_0$ (reference group)
2. $\bar{X}_B = \hat{\beta}_0 + \hat{\beta}_1$ (first level)
3. $\bar{X}_C = \hat{\beta}_0 + \hat{\beta}_2$ (second level)

If we knew the true mean grades in each class (μ_A, μ_B, μ_C, respectively), how could we obtain the model parameters directly?

1. $\beta_0 = \mu_A$: average grade in the reference group (Class A)
2. $\beta_1 = \mu_B - \mu_A$: difference in average grades of Classes B and A

3. $\beta_2 = \mu_C - \mu_A$: difference in average grades of Classes C and A

#Model 2: deviation coding Whereas reference cell coding compared each level to a reference level, deviation coding compares each level to a grand mean.

##Fit the model

```
options(contrasts = c("contr.sum", "contr.poly"))
mod <- lm(grades ~ classes)
summary(mod)
```

```
##
## Call:
## lm(formula = grades ~ classes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8958 -2.7384 -0.2568  2.3932 10.9884
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  74.3547     0.8478  87.698   <2e-16 ***
## classes1     -0.4471     1.1563  -0.387    0.702
## classes2      0.7542     1.3001   0.580    0.567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.517 on 27 degrees of freedom
## Multiple R-squared:  0.01245,    Adjusted R-squared:  -0.0607
## F-statistic: 0.1702 on 2 and 27 DF,  p-value: 0.8444
```

What does your $\boldsymbol{\beta}$ vector look like for this model?

$$\boldsymbol{\beta} = \begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{bmatrix}$$

Here, μ is the grand mean... but what's the grand mean anyway?

```
# is it the overall mean?
mean(grades)
```

```
## [1] 74.24391
```

```
# or is it the mean of the means of the dependent variable at all levels of
# the factor
class_means <- aggregate(grades ~ classes, FUN = mean) %>% data.frame()
mean(class_means$grades)
```

```
## [1] 74.35471
```

##Design matrix Obtain the design matrix for this coding scheme (ignoring replicates) from your `mod` object.

```
# save the unique rows of your design matrix
X = unique(model.matrix(mod))
X
```

```
##   (Intercept) classes1 classes2
## 1           1       -1       -1
## 2           1        0        1
```

```
## 5           1         1         0
```

How do we interpret the rows of `X`?

1. Class C is never compared to the other levels (because we obtain information about it from rows 1-2)
2. Class A is compared to all others
3. Class B is compared to all others

##Group means We can obtain the class-specific means from the model parameter estimates by multiplying the unique rows of the design matrix (`X`) by the coefficients.

```
X %*% mod$coefficients
```

```
##       [,1]
## 1 74.04761
## 2 75.10895
## 5 73.90756
```

How do these means compare to those we calculated initially using `aggregate()`?

Still the same.

So what does this mean? When using deviation coding, we have

1. $\bar{X}_A = \hat{\mu} + \hat{\alpha}_1$
2. $\bar{X}_B = \hat{\mu} + \hat{\alpha}_2$
3. $\bar{X}_C = \hat{\mu} - \hat{\alpha}_1 - \hat{\alpha}_2$

If we knew the true mean grades in each class (μ_A, μ_B, μ_C, respectively) and the true grand mean ($\bar{\mu}$), how could we obtain the model parameters directly?

1. $\mu = \bar{\mu}$: the mean of the class-specific mean grades
2. $\alpha_1 = \mu_1 - \bar{\mu}$: deviation of Class A's mean grade from the grand mean grade
3. $\alpha_2 = \mu_2 - \bar{\mu}$: deviation of Class B's mean grade from the grand mean grade

#Model 3: Helmert coding Whereas reference cell coding compared each level to a reference level and deviation coding compares each level to a grand mean, Helmert coding compares each level with the mean of the subsequent levels.

##Fit the model

```
options(contrasts = c("contr.helmert", "contr.poly"))
mod <- lm(grades ~ classes)
summary(mod)
```

```
##
## Call:
## lm(formula = grades ~ classes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8958 -2.7384 -0.2568  2.3932 10.9884
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  74.3547     0.8478  87.698   <2e-16 ***
## classes1      0.6007     1.0919   0.550    0.587
## classes2     -0.1535     0.5669  -0.271    0.789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.517 on 27 degrees of freedom
## Multiple R-squared:  0.01245,    Adjusted R-squared:  -0.0607
## F-statistic: 0.1702 on 2 and 27 DF,  p-value: 0.8444
```

What does your $\boldsymbol{\beta}$ vector look like for this model?

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

##Design matrix Obtain the design matrix for this coding scheme (ignoring replicates) from your `mod` object.

```
# save the unique rows of your design matrix
X = unique(model.matrix(mod))
X
```

```
##   (Intercept) classes1 classes2
## 1           1        0        2
## 2           1        1       -1
## 5           1       -1       -1
```

How do we interpret the rows of `X`?

1. Class A is compared to Classes B and C
2. Class B is compared to Class C
3. Class C is never compared to the other levels (because we obtain information about it from rows 1-2)

##Group means We can obtain the class-specific means from the model parameter estimates by multiplying the unique rows of the design matrix (`X`) by the coefficients.

```
X %*% mod$coefficients
```

```
##       [,1]
## 1 74.04761
## 2 75.10895
## 5 73.90756
```

How do these means compare to those we calculated initially using `aggregate()`?

> Still the same.

So what does this mean? When using deviation coding, we have

1. $\bar{X}_A = \hat{\beta}_0 - \hat{\beta}_1 - \hat{\beta}_2$
2. $\bar{X}_B = \hat{\beta}_0 + \hat{\beta}_1 - \hat{\beta}_2$
3. $\bar{X}_C = \hat{\beta}_0 + 2\hat{\beta}_2$

If we knew the true mean grades in each class (μ_A, μ_B, μ_C, respectively) and the true grand mean ($\bar{\mu}$), how could we obtain the model parameters directly? Interpret.

1. $\beta_0 = \bar{\mu}$: the mean of the class-specific mean grades
2. $\beta_1 = \frac{1}{2}(\mu_2 - \mu_1)$: the average difference between the means of Classes A and B
3. $\beta_2 = \frac{1}{3}(\mu_3 - \frac{1}{2}(\mu_2 - \mu_1)) = \frac{1}{3options}(\mu_3 - \beta_1)$: the average difference between the mean of Class C and the average difference between the means of Classes A and B