Tutorial 6: Dummy Variables and Interactions

Anh Le (anh.le@duke.edu)

Oct 2, 2015

Agenda

1. Merging data
2. Factor
3. Regression with factors
4. Regression with interaction term
5. Tips & Tricks

- the etymology of dummy vs binary
- Latex location syncing between pdf and source code
- Google R style guide https://google-styleguide.googlecode.com/svn/trunk/Rguide.xml

1. Merging data

Walk through syntax examples in http://www.princeton.edu/~otorres/Merge101R.pdf

Extended practice in homework

2. Factor

Factor is an `R` data type to encode categorical data. (Factor is a `R` term, categorical data is a Statistics term).

In the real world, categorical data come in two forms: numeric and characters.

Consider when the raw data is numeric

`schtyp` refers to school type, with 0 = private and 1 = public

```
set.seed(124)
schtyp <- sample(0:1, 20, replace = TRUE)
schtyp
```

```
##  [1] 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0
```

```
is.factor(schtyp)
```

```
## [1] FALSE
```

```r
is.numeric(schtyp)
```

```
## [1] TRUE
```

We convert the raw data into a factor as follows:

```r
schtyp.f1 <- factor(schtyp)
schtyp.f1
```

```
##  [1] 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0
## Levels: 0 1
```

```r
is.factor(schtyp.f1)
```

```
## [1] TRUE
```

```r
# Note how we also supply the labels
schtyp.f2 <- factor(schtyp,
                    levels = c(0, 1),
                    labels = c("private", "public"))
schtyp.f2
```

```
##  [1] private private public  private private private public  private
##  [9] public  private public  public  public  public  private private
## [17] public  public  public  private
## Levels: private public
```

```r
is.factor(schtyp.f2)
```

```
## [1] TRUE
```

Consider when the raw data is character.

`ses` referes to social economic status.

```r
ses <- c("low", "middle", "low", "low", "low", "low", "middle", "low", "middle",
    "middle", "middle", "middle", "middle", "high", "high", "low", "middle",
    "middle", "low", "high")
is.factor(ses)
```

```
## [1] FALSE
```

```r
is.character(ses)
```

```
## [1] TRUE
```

```r
factor(ses)
```

```
##  [1] low    middle low    low    low    low    middle low    middle middle
## [11] middle middle middle high   high   low    middle middle low    high
## Levels: high low middle
```

Note how the levels are arranged alphabetically and may not correspond to "real-world" order. Whether this is a problem depends on whether you want to model your data as categorical or ordinal. Usually, you want to model your data as categorical, as `R` treats ordinal variable strangely.

3. Regression with dummies

```r
# Load the LDC_IO dataset that you used for homework
library(foreign)
d <- read.dta('LDC_IO_replication.dta')
```

Regression with factors

`aclpn` is a binary variable, with 1 = democracy

```r
summary(lm(fdignp ~ factor(aclpn), data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ factor(aclpn), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.736  -1.501  -1.114   0.219 183.064
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.5007     0.1189  12.623  < 2e-16 ***
## factor(aclpn)1    1.1995     0.2023   5.928 3.41e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.293 on 3025 degrees of freedom
##   (2343 observations deleted due to missingness)
## Multiple R-squared:  0.01148,    Adjusted R-squared:  0.01116
## F-statistic: 35.14 on 1 and 3025 DF,  p-value: 3.408e-09
```

Regression with multiple dummies to encode a categorical variable

`dictator1` is a categorical variable that denotes the regime type. For example, 1: single party, . . . , 7: sp/pers/mil, 8: democracy

If we put it straight in the regression, what would be the problem?

```r
summary(lm(fdignp ~ bpc1 + dictator1, data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + dictator1, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.769  -1.720  -1.123   0.245 182.865
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.79093    0.29201   2.709  0.00681 **
## bpc1         0.57620    0.24301   2.371  0.01782 *
## dictator1    0.16626    0.04191   3.967  7.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.554 on 2189 degrees of freedom
##   (3178 observations deleted due to missingness)
## Multiple R-squared:  0.009206,   Adjusted R-squared:  0.008301
## F-statistic: 10.17 on 2 and 2189 DF,  p-value: 4.017e-05
```

So R mistakenly thought that dictator1 is a continuous variable above. We want to model dictator1 as a categorical variable instead.

```r
summary(lm(fdignp ~ bpc1 + factor(dictator1), data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + factor(dictator1), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.848  -1.992  -0.764   0.432 181.544
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.1573     0.3730  -0.422  0.67328
## bpc1                 0.7701     0.2460   3.130  0.00177 **
## factor(dictator1)2   2.4076     0.4164   5.782 8.43e-09 ***
## factor(dictator1)3   0.4589     0.5980   0.767  0.44296
## factor(dictator1)4   0.3890     1.0400   0.374  0.70837
## factor(dictator1)5   0.4774     0.5801   0.823  0.41059
## factor(dictator1)6   1.1890     0.7493   1.587  0.11270
## factor(dictator1)7   1.3795     0.4510   3.059  0.00225 **
## factor(dictator1)8   2.4774     0.3811   6.501 9.87e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.499 on 2183 degrees of freedom
##   (3178 observations deleted due to missingness)
## Multiple R-squared:  0.03137,    Adjusted R-squared:  0.02782
## F-statistic: 8.838 on 8 and 2183 DF,  p-value: 5.685e-12
```

The output will look a lot better with label. (Quiz: What's the average FDI for democracy?)

```
d$dictator1 <- factor(d$dictator1,
                      levels = c(1, 2, 3, 4, 5, 6, 7, 8),
                      labels = c('sp', 'pers', 'mil',
                                 'sp/pers', 'pers/mil', 'sp/mil',
                                 'sp/pers/mil', 'democracy'))
summary(lm(fdignp ~ bpc1 + dictator1, data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + dictator1, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.848  -1.992  -0.764   0.432 181.544
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -0.1573     0.3730  -0.422  0.67328
## bpc1                    0.7701     0.2460   3.130  0.00177 **
## dictator1pers           2.4076     0.4164   5.782 8.43e-09 ***
## dictator1mil            0.4589     0.5980   0.767  0.44296
## dictator1sp/pers        0.3890     1.0400   0.374  0.70837
## dictator1pers/mil       0.4774     0.5801   0.823  0.41059
## dictator1sp/mil         1.1890     0.7493   1.587  0.11270
## dictator1sp/pers/mil    1.3795     0.4510   3.059  0.00225 **
## dictator1democracy      2.4774     0.3811   6.501 9.87e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.499 on 2183 degrees of freedom
##   (3178 observations deleted due to missingness)
## Multiple R-squared:  0.03137,    Adjusted R-squared:  0.02782
## F-statistic: 8.838 on 8 and 2183 DF,  p-value: 5.685e-12
```

Note that a level is missing. Why?

At this point, the coefficient is a comparison of all categories against `dictator1 = 1 (sp)`. This may not be a good category to base our comparison against.

Theoretically, we want to compare all types of dictatorships against democracy instead. In R, If we want to choose another category to compare against, we need to change the "reference level" of the factor variable as follows.

```
# Change the reference level to 'democracy'
d$dictator1 <- relevel(d$dictator1, ref = 'democracy')

# Note how democracy is now the reference level, and thus omitted
summary(lm(fdignp ~ bpc1 + dictator1, data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + dictator1, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.848  -1.992  -0.764   0.432 181.544
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.32008    0.23723   9.780  < 2e-16 ***
## bpc1                    0.77007    0.24602   3.130 0.001771 **
## dictator1sp            -2.47736    0.38108  -6.501 9.87e-11 ***
## dictator1pers          -0.06975    0.32355  -0.216 0.829340
## dictator1mil           -2.01849    0.53892  -3.745 0.000185 ***
## dictator1sp/pers       -2.08831    1.01133  -2.065 0.039049 *
## dictator1pers/mil      -1.99992    0.51565  -3.878 0.000108 ***
## dictator1sp/mil        -1.28841    0.70008  -1.840 0.065850 .
## dictator1sp/pers/mil -1.09782    0.37392  -2.936 0.003360 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.499 on 2183 degrees of freedom
##   (3178 observations deleted due to missingness)
## Multiple R-squared:  0.03137,    Adjusted R-squared:  0.02782
## F-statistic: 8.838 on 8 and 2183 DF,  p-value: 5.685e-12
```

Food for thought: What to do when the categorical variable has a lot of levels?

`polityiv_update2` has 20 levels, ranging from -10 (most undemocratic) to 10 (most democratic). What to do?

```
summary(lm(fdignp ~ bpc1 + factor(polityiv_update2), data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + factor(polityiv_update2), data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.186  -1.213  -0.629   0.294 179.494
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.85446    0.78116   3.654 0.000265 ***
## bpc1                          0.11589    0.27029   0.429 0.668152
## factor(polityiv_update2)-9 -2.04884    0.90817  -2.256 0.024187 *
## factor(polityiv_update2)-8 -2.30724    0.92493  -2.495 0.012700 *
## factor(polityiv_update2)-7 -2.01978    0.81685  -2.473 0.013502 *
## factor(polityiv_update2)-6 -1.54186    0.93276  -1.653 0.098499 .
## factor(polityiv_update2)-5  2.10080    0.99785   2.105 0.035399 *
## factor(polityiv_update2)-4 -2.00090    1.11688  -1.792 0.073375 .
## factor(polityiv_update2)-3 -1.42872    1.08823  -1.313 0.189384
## factor(polityiv_update2)-2 -2.32527    1.22715  -1.895 0.058268 .
## factor(polityiv_update2)-1 -2.07329    1.16339  -1.782 0.074897 .
## factor(polityiv_update2)0  -1.81669    1.90068  -0.956 0.339295
## factor(polityiv_update2)1  -0.07971    1.44878  -0.055 0.956128
## factor(polityiv_update2)2  -1.73993    1.57435  -1.105 0.269229
## factor(polityiv_update2)3  -1.28779    1.37935  -0.934 0.350622
## factor(polityiv_update2)4  -0.13309    1.16877  -0.114 0.909350
## factor(polityiv_update2)5  -1.59560    1.01460  -1.573 0.115972
## factor(polityiv_update2)6  -0.51165    0.93950  -0.545 0.586095
## factor(polityiv_update2)7  -1.91454    0.95330  -2.008 0.044754 *
## factor(polityiv_update2)8  -0.92858    0.87708  -1.059 0.289863
## factor(polityiv_update2)9  -0.67664    0.88504  -0.765 0.444648
## factor(polityiv_update2)10 -0.94278    0.94172  -1.001 0.316898
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.491 on 1844 degrees of freedom
##   (3504 observations deleted due to missingness)
## Multiple R-squared:  0.03004,    Adjusted R-squared:  0.01899
## F-statistic: 2.719 on 21 and 1844 DF,  p-value: 4.215e-05
```

```r
summary(lm(fdignp ~ bpc1 + polityiv_update2, data = d))
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + polityiv_update2, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.579  -1.346  -0.894   0.202 183.133
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.57796    0.20008   7.886 5.25e-15 ***
## bpc1              0.07406    0.26630   0.278   0.7810
## polityiv_update2  0.04414    0.01844   2.393   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.538 on 1863 degrees of freedom
##   (3504 observations deleted due to missingness)
## Multiple R-squared:  0.003095,   Adjusted R-squared:  0.002024
## F-statistic: 2.892 on 2 and 1863 DF,  p-value: 0.05573
```

This is no longer a statistical issue but a theoretical / substantive question. On the one hand, `polityiv_update2` has a lot of levels, so it kinda looks more like a continuous variable rather than 20 categories. On the other hand, if we do so, there's only one coefficient for the variable `polityiv_update2`, meaning that for each 1 unit increase in `polityiv_update1`, `fdignp` increases by that same amount. Are you comfortable with considering all one-unit changes in `polityiv` (e.g. from -10 to -9, 0 to 1, or 9 to 10) as equivalent?

It's a tricky question. (I guess that's why a lot of people don't use Polity IV anymore).

Regression with interaction term

bpc1 is a binary variable, with 1 = having a balance of payment crisis aclpn is a binary variable, with 1 = democracy

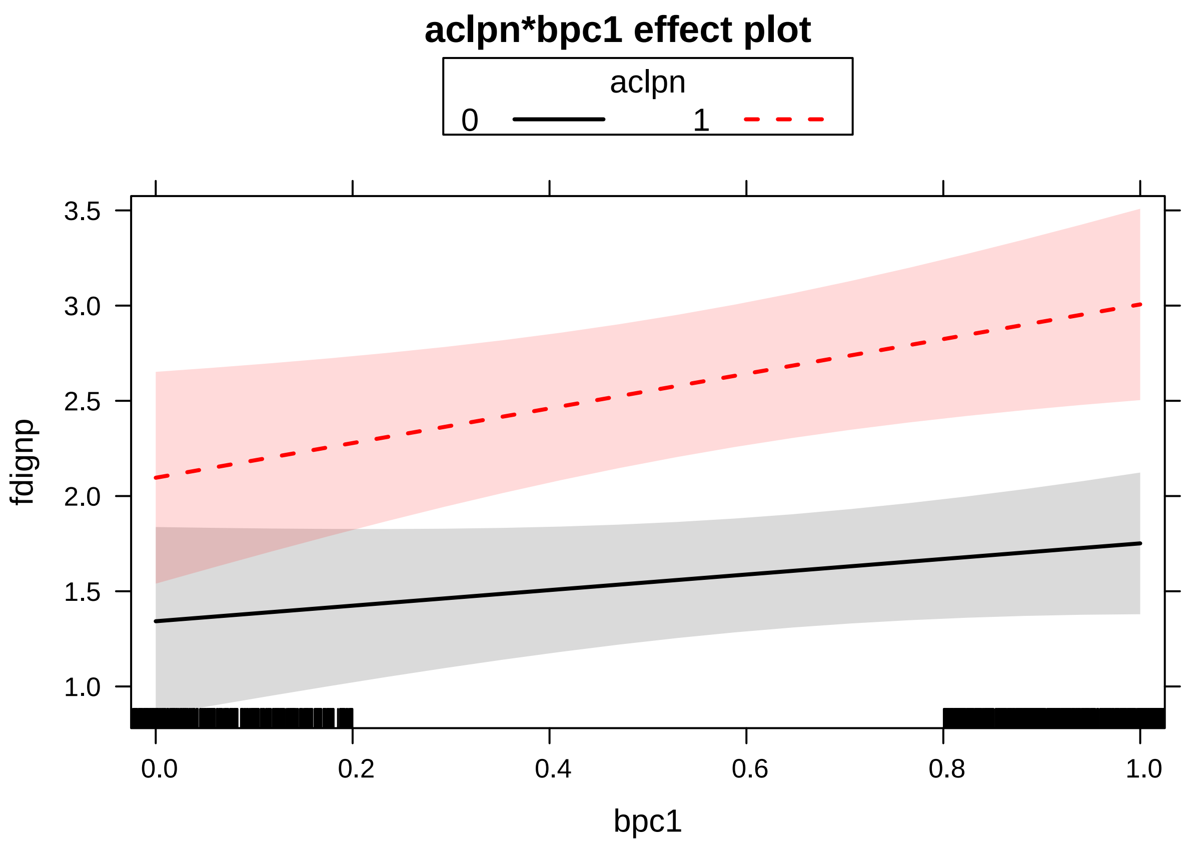
```r
d$aclpn <- factor(d$aclpn)
m_interaction <- lm(fdignp ~ bpc1 + aclpn + bpc1 * aclpn, data = d)
summary(m_interaction)
```

```
##
## Call:
## lm(formula = fdignp ~ bpc1 + aclpn + bpc1 * aclpn, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.987  -1.751  -1.151   0.248 182.813
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.3424     0.2523   5.321 1.14e-07 ***
## bpc1           0.4089     0.3157   1.296   0.1953
## aclpn1         0.7537     0.3796   1.986   0.0472 *
## bpc1:aclpn1    0.5012     0.4957   1.011   0.3121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.55 on 2188 degrees of freedom
##   (3178 observations deleted due to missingness)
## Multiple R-squared:  0.01087,    Adjusted R-squared:  0.009509
## F-statistic: 8.012 on 3 and 2188 DF,  p-value: 2.609e-05
```

Quiz: What's the FDI for a dictatorship undergoing BoP crisis? A democracy undergoing BoP crisis?

We can visualize the interaction effect as follows. (Next lab we will cover how to plot interaction in details.)

```r
library(effects)
plot(Effect(c("aclpn", "bpc1"), mod=m_interaction, se=TRUE),
     x.var = "bpc1",
     multiline=TRUE, ci.style = 'bands')
```


aclpn*bpc1 effect plot
aclpn
0
1
fdignp
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
bpc1