The Dummy Variable Trap with R

Econ 440 - Introduction to Econometrics

Patrick Toche, ptoche@fullerton.edu

12 May 2022

Create regions:

To illustrate regression with binary variables, let's group California counties by geographical area, based on Census data. To keep the example simple, let's **arbitrarily** create 4 regions named: North, South, East, West, based on a cursory look at a map.

For instance, here is what the list *North.Counties* looks like.

```
North.Counties
```

```
##  [1] "Butte"         "Colusa"        "El Dorado"     "Glenn"
##  [5] "Lassen"        "Modoc"         "Nevada"        "Placer"
##  [9] "Plumas"        "Sacramento"    "Shasta"        "Sierra"
## [13] "Siskiyou"      "Sutter"        "Tehama"        "Yolo"
## [17] "Yuba"          "Del Norte"     "Humboldt"      "Lake"
## [21] "Mendocino"     "Napa"          "Sonoma"        "Trinity"
## [25] "Alameda"       "Contra Costa"  "Marin"         "San Francisco"
## [29] "San Mateo"     "Santa Clara"   "Solano"        "Alpine"
## [33] "Amador"        "Calaveras"     "Madera"        "Mariposa"
## [37] "Merced"        "Mono"          "San Joaquin"   "Stanislaus"
## [41] "Tuolumne"
```

To view the complete code, see the `Rmd` file.

Create dummy variables for each region:

```
df$North <- df$county %in% North.Counties
df$East <- df$county %in% East.Counties
df$South <- df$county %in% South.Counties
df$West <- df$county %in% West.Counties
head(df[c("North", "East", "South", "West")], 10)
```

```
## # A tibble: 10 x 4
##    North East  South West
##    <lgl> <lgl> <lgl> <lgl>
##  1 TRUE  FALSE FALSE FALSE
##  2 TRUE  FALSE FALSE FALSE
##  3 TRUE  FALSE FALSE FALSE
##  4 TRUE  FALSE FALSE FALSE
##  5 TRUE  FALSE FALSE FALSE
##  6 FALSE TRUE  FALSE FALSE
##  7 TRUE  FALSE FALSE FALSE
##  8 FALSE TRUE  FALSE FALSE
##  9 FALSE TRUE  FALSE FALSE
## 10 TRUE  FALSE FALSE FALSE
```

Check consistency of the categories:

```
unique(df$North + df$East + df$South + df$West) == 1
```

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

Create a categorical variable for all 4 regions:

```
df$Region <- NA
dummies <- c("North", "East", "South", "West")
for (col in dummies)
  df$Region[which(df[,col] == TRUE)] <- col
head(df$Region, 10)
```

```
##  [1] "North" "North" "North" "North" "North" "East"  "North" "East"  "East"
## [10] "North"
```

Regression With Categorical Variables

If you estimate a linear regression with all categories and an intercept, R will automatically drop one of the categories:

```
m1 <- lm(TestScore ~ STR + Region, data=df)
summary(m1)
```

```
##
## Call:
## lm(formula = TestScore ~ STR + Region, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52.03 -13.10  -0.82  12.64  45.42
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  671.079      9.965   67.34  < 2e-16 ***
## STR           -1.558      0.488   -3.20   0.0015 **
## RegionNorth   18.081      2.393    7.56  2.7e-13 ***
## RegionSouth   11.892      2.861    4.16  3.9e-05 ***
## RegionWest    17.054      3.461    4.93  1.2e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.4 on 415 degrees of freedom
## Multiple R-squared:  0.17,   Adjusted R-squared:  0.162
## F-statistic: 21.2 on 4 and 415 DF,  p-value: 6.38e-16
```

The categorical variable *Region* stores the categories as strings (aka characters). When running the regression, R temporarily creates factor variables using alphabetical ordering, thus creating categories for $RegionNorth$, $RegionSouth$ and $RegionWest$, while omitting the first category in the list, $RegionEast$.

To use, say, $North$, as the reference region, create factors in the desired order, and run the regression again:

```
df$Region2 <- factor(df$Region, levels=c("North", "East", "South", "West"))
m2 <- lm(TestScore ~ STR + Region2, data=df)
summary(m2)
```

```
##
## Call:
## lm(formula = TestScore ~ STR + Region2, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52.03 -13.10  -0.82  12.64  45.42
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    689.160      9.349   73.72  < 2e-16 ***
## STR             -1.558      0.488   -3.20   0.0015 **
## Region2East    -18.081      2.393   -7.56  2.7e-13 ***
## Region2South    -6.189      2.442   -2.53   0.0116 *
## Region2West     -1.027      3.072   -0.33   0.7383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.4 on 415 degrees of freedom
## Multiple R-squared:  0.17,   Adjusted R-squared:  0.162
## F-statistic: 21.2 on 4 and 415 DF,  p-value: 6.38e-16
```

If the categories are already stored as factors but do not appear in the desired order, you can reset the factor levels:

```
df$Region3 <- relevel(df$Region2, ref="South")
m3 <- lm(TestScore ~ STR + Region3, data=df)
summary(m3)
```

```
##
## Call:
## lm(formula = TestScore ~ STR + Region3, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52.03 -13.10  -0.82  12.64  45.42
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    682.971     10.349   65.99  < 2e-16 ***
## STR             -1.558      0.488   -3.20   0.0015 **
## Region3North     6.189      2.442    2.53   0.0116 *
## Region3East    -11.892      2.861   -4.16  3.9e-05 ***
## Region3West      5.162      3.430    1.50   0.1331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.4 on 415 degrees of freedom
## Multiple R-squared:  0.17,   Adjusted R-squared:  0.162
## F-statistic: 21.2 on 4 and 415 DF,  p-value: 6.38e-16
```