STAT350 Tutorial 9

Dylan Maciel

3/11/2020

In this week's tutorial we'll be going over the use of indicator variables. Also referred to as dummy variables, these are used to incorporate categorical predictor variables into the linear regression model. Their typical form is:

$$x = \begin{cases} 1 \text{ if the variable is in a specific category,} \\ 0 \text{ if the variable is in the other category(s).} \end{cases}$$

If we think about this in the context of regression, inclusion of this variable leads to a difference in the intercept coefficient for the categories. So, consider a SLR with one dummy variable for a binary predictor:

$$y = \beta_0 + \beta_1 x.$$

The result is a different constant given the value of x; if $x = 0$, $y = \beta_0$ and if $x = 1$, $y = (\beta_0 + \beta_1)$. If a variable has a levels it will need $a - 1$ dummy variables. As the number of categorical variables increases and the number of levels for each variable increases, the model specified will become more complex.

Another way to introduce complexity into the model is to add interaction terms between predictor variables. The simplest example of which is an interaction between a continuous predictor $x_1$and a binary predictor x_2. The model we'll specify in this case is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2.$$

Here the fourth term is the interaction term. And, similar to the intercept, the value of the slope coefficient can change depending of the level of the binary predictor. Specifically, we get two possible equations.

$$y = \begin{cases} \beta_0 + \beta_1 x_1 & \text{if } x_2 = 0, \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3) x_1 & \text{if} x_2 = 1. \end{cases}$$

With all these possible models, we'll need to introduce some hypothesis tests. Derek went over these in lecture: a test for a difference in intercepts (parallel lines), a test for difference in slopes (concurrent lines).

and a test for the difference in the intercepts and slopes (coincident lines). For all of these we have a test statistic of the form

$$F = \frac{[SS_{Res}(RM) - SS_{Res}(FM)]/(df_{RM} - dfFM)}{MS_{Res}(FM)};$$

comparing the sum of squared residuals of the reduced model to the sum of squared residuals of the full model.

An Example

For our example we’ll again take a dataset from the `faraway` package. The `sexab` data consists of data from a study done on women who suffer from post-traumatic stress disorder (`ptsd`) with reported childhood sexual abuse (`csa`) and childhood physical abuse (`cpa`). `csa` is a binary variable while the ither two are continuous.

```r
library(faraway)
data(sexab)
str(sexab)
```

```
## ’data.frame':    76 obs. of  3 variables:
##  $ cpa : num  2.048 0.839 -0.241 -1.115 2.015 ...
##  $ ptsd: num  9.71 6.17 15.16 11.31 9.95 ...
##  $ csa : Factor w/ 2 levels "Abused","NotAbused": 1 1 1 1 1 1 1 1 1 1 ...
```

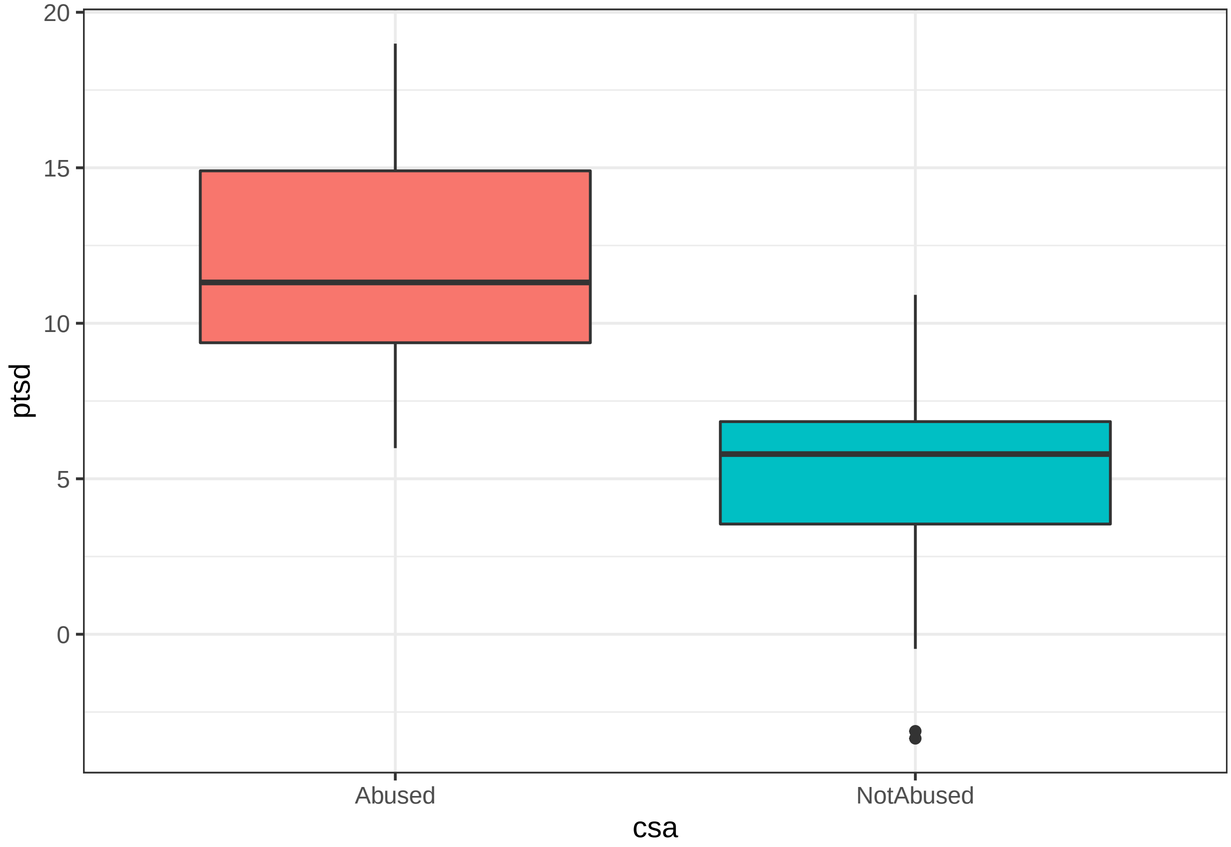
One thing to notice is that `csa` is coded as a factor, if it wasn’t we would have to change it to this format using the `as.factor()` function before our analysis.

It’s always a good idea to do some initial EDA before analysis; maybe we can find a visual difference between the two groups visually. I made the following plots with `ggplot`.

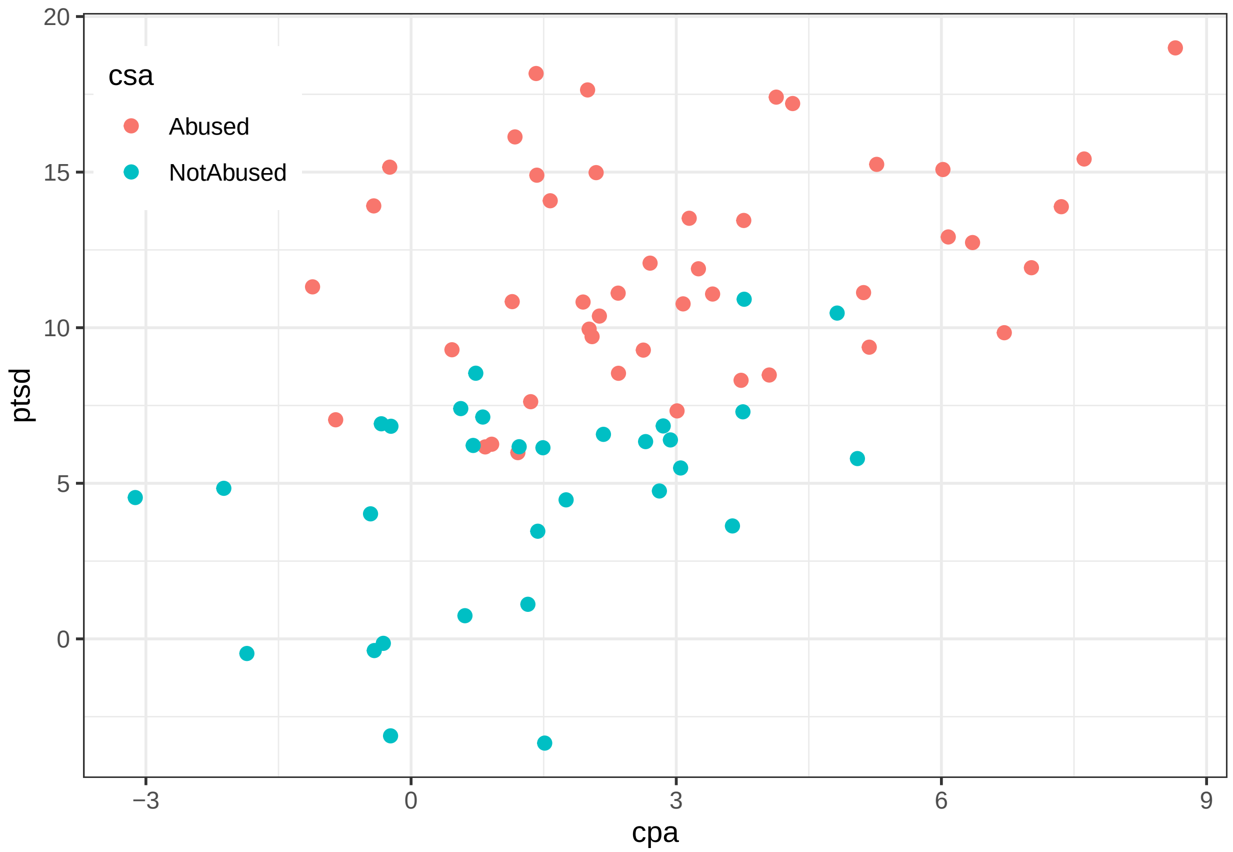
First, we have boxplots for `ptsd` given fir the levels of `csa`. Here there seems to be a difference between the two groups; we’ll test this hypothesis later.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```



Next, we have a scatter plot for `ptsd` given `cpa` where we have set the colour to indicate the level of `csa`.



Here there does seem to be a weak linear relationship between `cpa` and `ptsd` and there we also see a stratification where those who were abused have higher levels of `ptsd`. You may remember from previous classes that you can test the hypothesis for a difference using a t-test.

Next I'll show you how to code dummy variables. We don't actually need to do tis for our dataset as there is only one binary predictor, but you may need this for future assignments. I'll code two dummy variables, one for those who were sexually abused, and one for those not abused. Here I'll be using the `ifelse()` function to assign new values manually. There are functions from packages that can do this automatically, like `dummyVars` from the `caret` package, that are useful when the number of dummy varibles you'll have is large.

```
dum1 <- ifelse(sexab$csa == 'Abused', yes = 1, no = 0)
dum2 <- ifelse(sexab$csa == 'NotAbused', yes = 1, no = 0)
```

IF you recall the model specification from above, the model where $x = 0$ for the dummy variable is known as the reference level. The choice for this is arbitrary, but it is good practice to choose a reference level that makes sense in the context of the data. For this data the no abuse is the natural choice so I will make sure that `R` uses this a the reference level through the `relevel()` function.

```
sexab$csa <- relevel(sexab$csa, ref = ’NotAbused’)
```

Once we have are variables definde we can move onto fitting the model. First, we’ll fit the full model with interaction. Notice on the right of the ~ I have put `cpa*csa` and the model includes four coefficients.

```
mdl_full <- lm(ptsd ~ cpa*csa, data = sexab)
summary(mdl_full)
```

```
##
## Call:
## lm(formula = ptsd ~ cpa * csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1999 -2.5313 -0.1807  2.7744  6.9748
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6959     0.7107   5.201 1.79e-06 ***
## cpa            0.7640     0.3038   2.515   0.0142 *
## csaAbused      6.8612     1.0747   6.384 1.48e-08 ***
## cpa:csaAbused -0.3140     0.3685  -0.852   0.3970
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.279 on 72 degrees of freedom
## Multiple R-squared:  0.5828, Adjusted R-squared:  0.5654
## F-statistic: 33.53 on 3 and 72 DF,  p-value: 1.133e-13
```

The first thing to notice in the model summary is that the interaction term is insignificant, i.e. we reject the hypo. So, we’ll fit a model without it.

```
mdl_add <- lm(ptsd ~ cpa + csa, data = sexab)
summary(mdl_add)
```

```
##
## Call:
```

```
## lm(formula = ptsd ~ cpa + csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1567 -2.3643 -0.1533  2.1466  7.1417
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9753     0.6293   6.317 1.87e-08 ***
## cpa           0.5506     0.1716   3.209  0.00198 **
## csaAbused     6.2728     0.8219   7.632 6.91e-11 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.273 on 73 degrees of freedom
## Multiple R-squared:  0.5786, Adjusted R-squared:  0.5671
## F-statistic: 50.12 on 2 and 73 DF,  p-value: 2.002e-14
```

We can use an ANOVA to test the null hypothesis if that there is no difference between the above models. This is how you would get the F-statistic for comparing the full modelto any model nested within it.

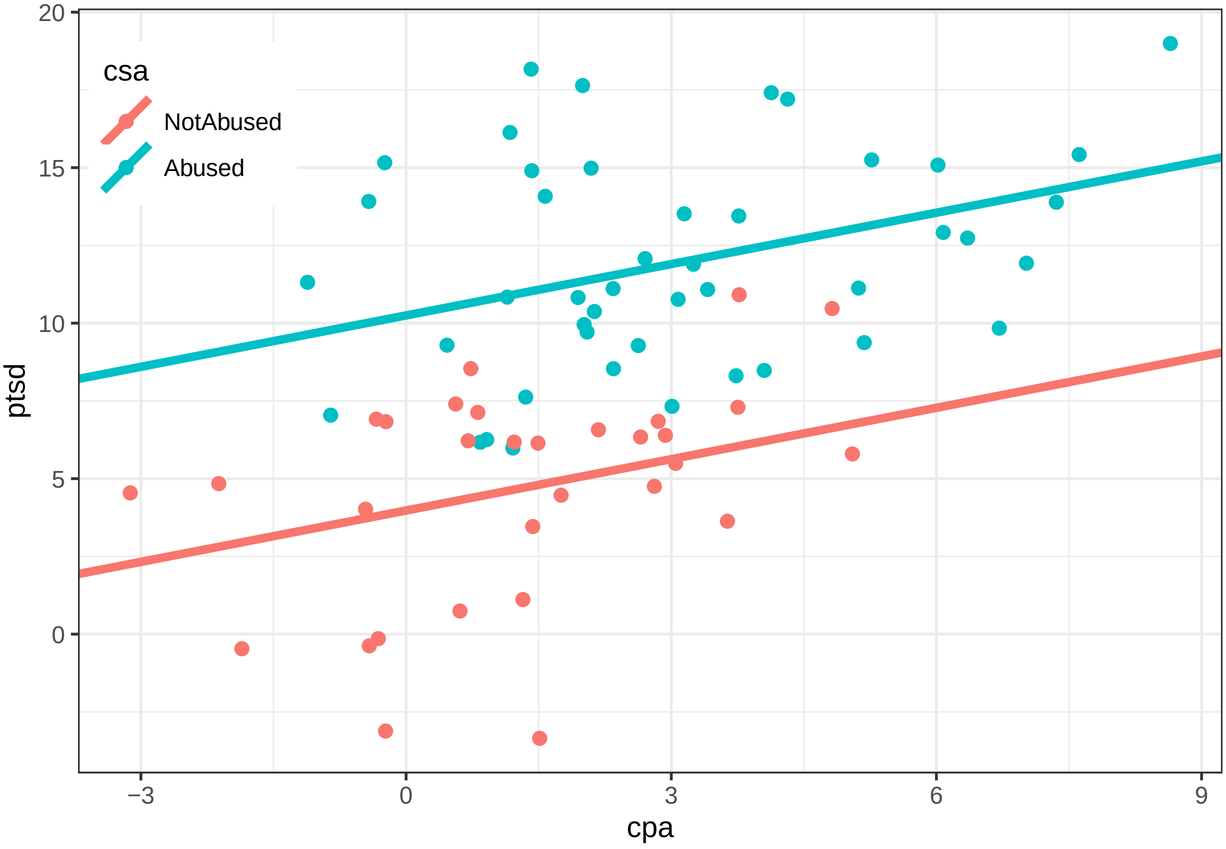
```
anova(mdl_full, mdl_add)
```

```
## Analysis of Variance Table
##
## Model 1: ptsd ~ cpa * csa
## Model 2: ptsd ~ cpa + csa
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     72 774.28
## 2     73 782.08 -1   -7.8069 0.726  0.397
```

So, we fail to reject the null and conclude that there is no difference.

We can plot the stratified regression lines from this model:

```
ggplot() +
  theme_bw() +
  theme(legend.position = c(0.1, 0.85)) +
  geom_point(aes(x = cpa, y = ptsd, col = csa), sexab, size = 2) +
  geom_abline(aes(slope = mdl_add$coefficients[2],
                  intercept = mdl_add$coefficients[1],
                  col = 'NotAbused'),
              size = 1.5) +
  geom_abline(aes(slope = mdl_add$coefficients[2],
                  intercept = mdl_add$coefficients[1] + mdl_add$coefficients[3],
                  col = 'Abused'),
              size = 1.5)
```


csa
NotAbused
Abused
ptsd
cpa