Adding Categorical Variables

Multiple linear regression – Stat 230

In this class, you will extend the multiple linear regression model to include categorical variables. You will consider how to do this, how to interpret the results, and how to determine whether the addition of a categorical variable discernibly improves the model (i.e., explains discernibly more variability in the response).

In this activity we will revisit the `Cars` data set described in Sections 3.1 and the first page of Section 3.3 (page 70).

```
cars <- read.csv("https://aloy.rbind.io/kuiper_data/cars.csv")
```

Adding a categorical variable

Task 1. (Activity 17) Make boxplots or individual value plots of `log(Price)` versus the categorical variables `Make`, `Model`, `Trim`, and `Type`.

Task 2. (Adaptation of Activity 18) Create five indicator variables that can be used to represent the six possible values of `Make` (Buick, Cadillac, Chevrolet, Pontiac, SAAB, Saturn). To answer this, just describe *how* you would define them, there is nothing you need to compute in R.

Task 3. (Adaptation of Activity 19) Build (fit) a multiple regression model using `log(Price)` as the response, and `Mileage` and the indicator variables for `Make`. To do this, you can simply add `Make` into the `lm()` statements we saw last class. R will automatically convert a categorical explanatory variable into a set of indicator variables. Report the fitted regression equation and the R^2 value. Which level of `Make` is the baseline. The baseline level is represented by 0's across all of the indicator variables.

Task 4. The inclusion of `Make` in the regression model allows each `Make` to have a different y-intercept but a common slope. Write the fitted regression equation for each `Make` (you will have six equations in total, all with the same slope). You can do this by hand if it takes a lot of time to type it in R Markdown.

Task 5. Now, let's plot the fitted "parallel slopes" model that you just wrote out. To do this, we can use the `ggpredict()` and `plot()` functions in the {ggeffects} package. Use this plot to verify your work in the previous task.

```
# Be sure to load ggeffects in the setup chunk!

# First, make predictions from the model
```

```
# Fill in the blank with your model name
cars_pred <- ggpredict(___, terms = ~Mileage + Make)

# Now plot the predictions
plot(cars_pred)
```

Tips on plotting the model

Holding other variables at "typical" values

If you have more predictor variables in your model, then variables not specified in the `terms` argument the `ggpredict()` function are set to a specific value. Quantitative variables are set to their mean. Categorical variables are set to the mode (the level with the most observations).

Labels

If you need to adjust your axis labels or title, then you will need to add a `labs()` layer to your plot. For example,

```
plot(cars_pred) +
  labs(x = "My new x label", y = "My new y label",
       title = "My new title")
```

Testing a categorical variable

Is there evidence that `Make` is associated with price after accounting for mileage? To answer this we need to use an extra sums of squares F-test.

Task 6. Write down the hypotheses for the sums of squares F-test that can be used to determine whether `Make` is associated with price after accounting for mileage. Remember that this F-test is comparing two nested models, so start by thinking about what the full model is and what the reduced model is.

Task 7. You have already fit the full model above. Fit the reduced model and use the `anova()` command to run the extra sums of squares F-test. State your conclusion to this F-test.

Note

To run an extra sums of squares F-test, you can fit the `full` and `reduced` models and then pass them into `anova()` in the order you see below.

```
anova(reduced, full)
```

One thing to note is that R reports the SSE (which it labels `RSS`) for each model, not the SSR.

Task 8. If you prefer to work with SSR you can run `anova()` on only the full model, but you need to be careful about the order you add the variables into your multiple linear regression model (to make your life easier). Refit your model (if needed) to add `Mileage` as the first explanatory variable and `Make` as the second explanatory variable. Then run `anova()` and only pass in your full model. You should see an ANOVA table with rows `Mileage`, `Make`, and `Residuals`, in that order. Verify that the F-statistic for `Make` is the same as the F-statistic calculated in the previous task.

Adding additional categorical variables.

Optional task. If you get done with the above tasks before the end of class, revisit the exploratory plots you created in Task 1 and expand your multiple linear regression model to include the other potentially useful categorical variables. Think about how they can be interpreted and determine whether they discernibly improve the model.