Zeros and Ones

Indicator variables

An indicator variable, also known as a dummy variable, is a variable used to represent membership in a specific category of a categorical variable. Using only the values 0 and 1, the indicator variable will indicate the presence of a particular attribute or membership in a category of interest with a 1, and use a 0 for everything else.

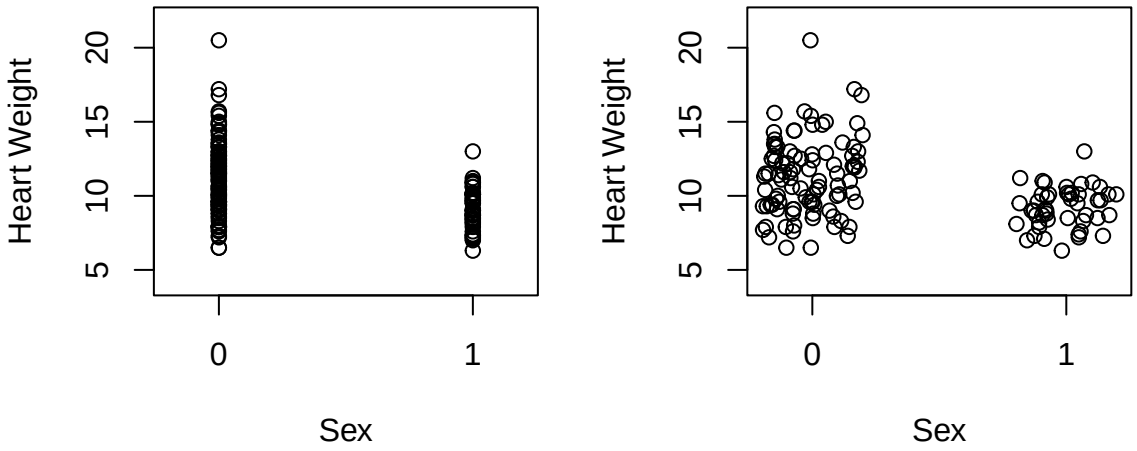
Indicator variables are essential when incorporating categorical variables—like treatment group, sex, or employment status—into linear regression models. For example, to represent whether each individual included in a survey is employed or unemployed, an indicator variable could be defined as 1 if the person is employed, and 0 if the person is unemployed.

For categorical variables with more than two categories, a separate indicator variable is created for each category except one, which becomes the reference (or baseline) group. This encoding, often called one-hot encoding, ensures that the model can estimate effects relative to the reference group while avoiding perfect collinearity (remember we need X^TX to have an inverse). For instance, if we instead treated employment as having three categories of full-time, part-time, or unemployed we would create two indicator variables: one for full-time and one for part-time. Unemployed would then serve as the reference group.

The choice of which group serves as the reference group is arbitrary, but it will affect the interpretation of model coefficients. For this reason, if your categorical variable is ordinal, it is customary to select either the lowest or the highest ordered category as your reference group; Interpretation of coefficients relative to max or min just often makes the most sense.

Indicator only Regression Model

While not commonly done, you can create a simple linear regression model using only an indicator variable as a predictor. Consider again the `cats` data frame in the `MASS` library. This data contains not only the heart weight and body weight of 144 cats as we've seen in earlier models, but also the sex of those 144 cats. From Figure 1 below it is clear that the heart weight of female cats is a bit less on average than the heart weight of male cats.



(a) Straight data

(b) Jittered sex for clarity

Figure 1: Cat heart weight by sex, 1=Female

If we fit a model of the form $y = \beta_0 + \beta_1 x + \varepsilon$ and have $x = 0$ for female cats and $x = 1$ for male cats what values of β_0 and β_1 should we expect? Since there are only two possible values of x, there will be only two possible fitted values from our model: one corresponding to when $x = 0$ and one for when $x = 1$. To minimize the sum of squared error, we want the output when $x = 0$ to be equal to the mean heart weight for female cats and the output when $x = 1$ to be equal to the mean heart weight for male cats. This is achieved when β_0 equals the mean heart weight for females and when $\beta_0 + \beta_1$ mean heart rate for males. Therefore β_1 should be the difference between the two group means.

Here's the work in R:

```
mean(cats$Hwt[cats$Sex=="F"])
```

```
[1] 9.202128
```

```
mean(cats$Hwt[cats$Sex=="M"])-mean(cats$Hwt[cats$Sex=="F"])
```

```
[1] 2.120553
```

```
mod_cat_sex<-lm(Hwt~Sex, data=cats)
summary(mod_cat_sex)$coef
```

```
            Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 9.202128  0.3250734 28.307842 2.959034e-60
SexM        2.120553  0.3960745  5.353924 3.379786e-07
```

Just as suspected, the result gives us an intercept equal to the mean heart weight of female cats and a slope equal to the difference between male and female heart weights.

Adding in an Indicator

You can also add an indicator term to a model to create a linear equation of the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

Continuing with the cat heart weight example, we could fit this type of model with y = heart weight, x_1 = body weight, and x_2 = sex (0=female, 1 = male)

The resulting model fit is:

```
               Estimate Std. Error    t value     Pr(>|t|)
(Intercept) -0.41495263  0.7273243 -0.5705194 5.692336e-01
Bwt          4.07576892  0.2947885 13.8260785 5.119676e-28
SexM        -0.08209684  0.3040474 -0.2700133 7.875448e-01
```

How can this be interpreted? First, note that the third row begins with `SexM` this is R's way of communicating that for the Sex variable level M is the one that is corresponding to the indicator equalling 1. This tells us that the heart weight of cats is, on average, equal to -0.415 plus (4.0758 $\times$ body weight) for female cats, and (-0.415 plus -0.0821) plus (4.0758 $\times$ body weight) for male cats. Same slope for all cats, but intercepts that differ by -0.0821.

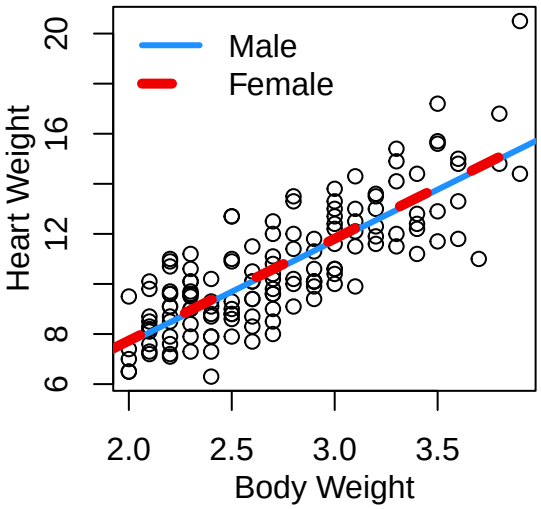


Figure 2: Cat heart weight as function of body weight

Hmmm… the two lines look pretty much the same. Why is that? Well take a look again at the model output above. We see that the difference in intercepts is only -0.0821, not a very big number. But scanning across this line in the output we also see that the p-value in the t-test of $H_o : \beta_2 = 0$ is 0.788 which means that with any reasonable α level, we would fail to reject the null. This means the coefficient associated with sex is not significantly different from zero, and the two lines aren't really needed - one will do just fine.

Other times the indicator will matter quite a bit. Consider the `mtcars` dataframe from the `datasets` library. In a model predicting vehicle fuel efficiency as a function of horse power, does it help if we add in an indicator that is 1 for a manual transmission and 0 for an automatic? Our model will again be in the form of:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

with y =mpg, x_1 =horsepower, and x_2 =manual transmission.

```
             Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 26.5849137 1.425094292 18.654845 1.073954e-17
hp          -0.0588878 0.007856745 -7.495191 2.920375e-08
am           5.2770853 1.079540576  4.888270 3.460318e-05
```

In this case, the test of $H_o : \beta_2 = 0$ has a p-value that is 3×10^{-5} meaning there is strong evidence this additional term related to transmission type matters. In a plot of the fitted model we see:

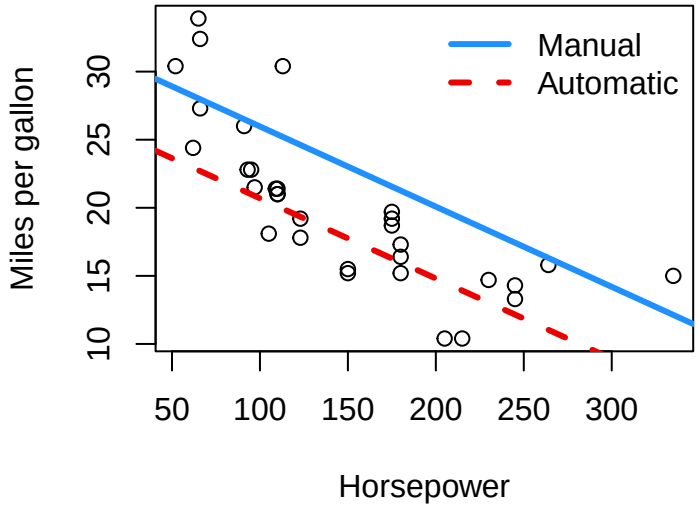


Figure 3: Car MPG as function of HP

Figure 3 shows us very clearly that cars of equal horsepower tend to get better gas mileage if they are a manual transmission. The difference between the two lines is β_2, 5.2771.

Interacting with an Indicator

Parallel lines are great and all, but there's no reason to believe that the best model fit for two levels of your indicator should always have the same slope. Ideally we want a model that takes a form more like:

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

With this structure, for our baseline level of $x_2 = 0$, the β_2 and β_3 terms disappear since they are multiplied by zero, leaving you with the basic β_0 as intercept and β_1 as slope. When $x_2 = 1$ though, the β_2 and β_3 terms stick around and $\beta_0 + \beta_2$ becomes the intercept and $\beta_1 + \beta_3$ becomes the slope. Having a $x_1 x_2$ term is called having an interaction between x_1 and x_2. The x_2 indicator value is interacting with the slope associated with your continuous x_1.

Revisiting the cat heart and body weight data, might sex play a significant role in predicting heart weight if we allow slope to change as well as intercept? Here's the R model summary for the revised model that includes the $\beta_3 x_1 x_2$ term:

```
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  2.981312  1.8428394  1.617782 0.1079604636
Bwt          2.636414  0.7759022  3.397869 0.0008845733
SexM        -4.165400  2.0617552 -2.020318 0.0452578391
Bwt:SexM     1.676265  0.8373255  2.001927 0.0472246471
```

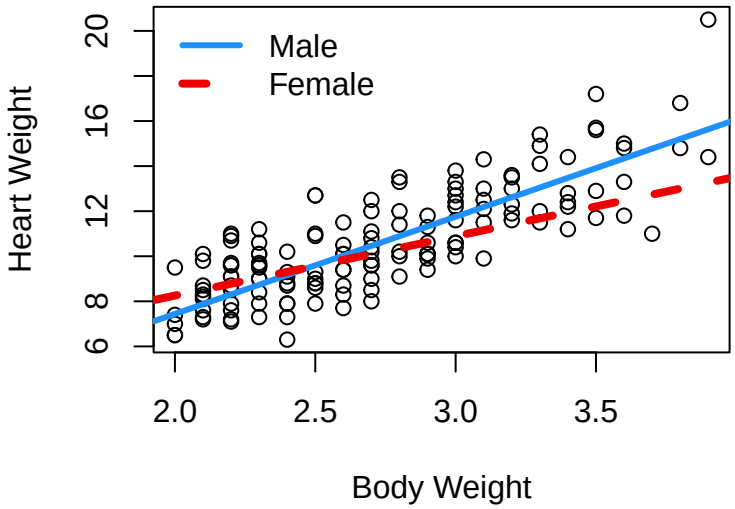


Figure 4: Cat heart weight as function of body weight

The last row labeled `Bwt:SexM` corresponds to our $\beta_3 x_1 x_2$ term. Reading across the output summary table shows us that this β_3 coefficient is estimated to be significantly different from 0 at the $\alpha = 0.05$ level. The β_2 term allowing for different intercepts that was not significant in the limited $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ model is now also showing a p-value below 0.05 at 0.0453. Plotted, the fits for male and female cats now look like the lines in Figure 4.

In R

In the above example using `mtcars`, you may have noticed that the reference level wasn't clear in the model summary output. That is because the transmission type variable, `am`, is pre-coded as a 0/1 indicator variable. The help menu for mtcars explains that am is a 0 for automatic, and 1 for manual. When read into R, it takes this 0/1 coding as a numeric 0/1 and doesn't think of it as a factor type needing a reference level at all. Mathematically that is fine - it just makes interpretation a little trickier because you need to remember what is coded as 0 and what is coded as a 1.

Here is the code for the model described earlier predicting `mpg` as a function of `horsepower` and transmission type (`am`) :

```
# adding on +am to mpg~hp creates an additive beta for transmission
mod_mpg<-lm(mpg~hp+am, data=mtcars)
summary(mod_mpg)$coef
```

```
              Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 26.5849137 1.425094292 18.654845 1.073954e-17
hp          -0.0588878 0.007856745 -7.495191 2.920375e-08
am           5.2770853 1.079540576  4.888270 3.460318e-05
```

A simple run of `?mtcars` to pull up the help menu will show you that `am=0` for automatics and therefore automatic is the baseline reference level for transmission type and the β_2 shown for `am` is the change in intercept associated with a manual transmission.

In the `cats` data set explored this chapter, `Sex` is coded with F and M levels and is recognized by R as a factor type variable. You can include factors in your `lm` function input formula as-is and R will take care of the 0/1 coding behind the scenes for you. The first level of your factor will be treated as the 0 reference level. Here is code that first looks at the levels of `Sex`, then creates and summarizes the cat heart weight model including indicator interaction:

```
# look at levels to know which will be reference group
levels(cats$Sex)
```

```
[1] "F" "M"
```

```r
# multiplying the indicator factor creates both an additive beta for
#   different intercepts and the beta for the slope change
mod_cats_full<-lm(Hwt~Bwt*Sex, data=cats)
summary(mod_cats_full)$coef
```

```
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  2.981312  1.8428394  1.617782 0.1079604636
Bwt          2.636414  0.7759022  3.397869 0.0008845733
SexM        -4.165400  2.0617552 -2.020318 0.0452578391
Bwt:SexM     1.676265  0.8373255  2.001927 0.0472246471
```

As pointed out earlier, the row labels of `SexM` and `Bwt:SexM` also make clear level M is the one that is being treated as a 1, meaning level F must be our 0 reference group. If you want that switched, you can use `relevel` to change the ordering of the levels of the factor variable.

Multi-level factors in Regression

For categorical variables with more than two categories, we create a separate indicator variable for all categories except one, our baseline reference group. The below illustrates this process with an example from the `Cars93` data set from the `MASS` library. Vehicle type is a factor that takes on six different values: Compact, Large, Midsize, Small, Sporty, and Van. To incorporate this one factor with six levels, we create five new variables: Large, Midsize, Small, Sporty, and Van; each coded as a 0 or a 1. We do not need a variable for Compact - the Compact car level is signaled by not being any of the other types. This is often called one-hot encoding.

Type
Small
Midsize
Compact
Midsize
Midsize
Sporty
Large
Small
Compact
Van
Large
Midsize

Large	Midsize	Small	Sporty	Van
0	0	1	0	0
0	1	0	0	0
0	0	0	0	0
0	1	0	0	0
0	1	0	0	0
0	0	0	1	0
1	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	1
1	0	0	0	0
0	1	0	0	0

Figure 5: One-hot encoding example

This is how car type can be included in a regression model - through the addition of five indicators. The order of the levels of car `Type` is, by default, alphabetical:

```
levels(Cars93$Type)
```

```
[1] "Compact" "Large"   "Midsize" "Small"   "Sporty"  "Van"
```

To build a model of vehicle highway mpg as a function of weight, allowing for different intercepts for each type, the code looks much like what we've seen with the `lm` command before:

```
mod_car93<-lm(MPG.highway~Weight+Type, data=Cars93)
summary(mod_car93)$coef
```

```
                Estimate  Std. Error     t value     Pr(>|t|)
(Intercept) 49.532032100 3.308855496 14.96953619 1.126489e-25
Weight      -0.006736186 0.001104706 -6.09772023 2.969242e-08
TypeLarge    2.088508950 1.450279960  1.44007296 1.534773e-01
TypeMidsize  0.098272245 1.115603939  0.08808883 9.300109e-01
TypeSmall    1.523993750 1.194802165  1.27551974 2.055600e-01
TypeSporty  -1.213784862 1.092190088 -1.11133115 2.695231e-01
TypeVan     -1.839809387 1.600556866 -1.14948080 2.535445e-01
```

The main difference here is that withe the simple `+Type` we've now added five more estimates; one for each of the new indicator terms. So the given intercept of 49.532 is the intercept for the baseline Compact car type, then we add 2.0885 to that to get the intercept for Large cars, add 0.0983 to 49.532 to get the intercept for Midsize cars, and so on through to adding -1.8398 to 49.532 for Vans.

Note in these results none of the p-values in the last column for these Type-related s indicate the is significantly different from 0. This does NOT mean that no two vehicle types have significantly different intercepts, it just means that none of the vehicle types have an intercept significantly different from Compact cars. If a different car type were selected to be the reference level, these estimates and their corresponding p-values would change. Below are the results from the same model but using Van as the baseline reference rather than Compact.

```
Type2<-relevel(Cars93$Type, "Van")
mod_car93_Van<-lm(MPG.highway~Weight+Type2, data=Cars93)
summary(mod_car93_Van)$coef
```

```
                Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)  47.692222713 4.346958390 10.9714008 5.017827e-18
Weight       -0.006736186 0.001104706 -6.0977202 2.969242e-08
Type2Compact  1.839809387 1.600556866  1.1494808 2.535445e-01
Type2Large    3.928318337 1.349446429  2.9110591 4.586149e-03
Type2Midsize  1.938081632 1.272889330  1.5225846 1.315314e-01
Type2Small    3.363803137 2.055312537  1.6366383 1.053606e-01
Type2Sporty   0.626024525 1.637942803  0.3822017 7.032546e-01
```

The resulting linear equations are exactly the same. Slope is obviously the same as before, and the intercepts are just expressed with relation to a different baseline. Intercept for Midsize cars for example was originally 49.532 + 0.0983=49.6303 and in the newer model the Midsize intercept is 47.6922 + 1.9381=49.6303. So no real change… just a different comparitor in the baseline position. You'll also see that with Van as the baseline, the associated with Large cars now has a p-value below 0.05 indicating it is significantly different from zero.

For the full model with interactions between car type and weight the addition of *Type leads to the addition of 10 new estimates: five for unique intercepts plus five for unique slopes. And just as we saw with the additive indicator model, the values of the estimates will change depending on what factor level is used as the baseline reference level.

With the default Compact as the reference level of type:

```
mod_car93_full<-lm(MPG.highway~Weight*Type, data=Cars93)
summary(mod_car93_full)$coef
```

```
                       Estimate   Std. Error    t value     Pr(>|t|)
(Intercept)         4.781871e+01 10.199629443  4.6882791 1.099859e-05
Weight             -6.149055e-03  0.003486294 -1.7637797 8.153978e-02
TypeLarge          -7.310074e+00 16.890945386 -0.4327807 6.663243e-01
TypeMidsize        -2.810940e+00 12.244349583 -0.2295704 8.190043e-01
TypeSmall           1.774275e+01 11.678767728  1.5192314 1.325981e-01
TypeSporty         -3.170950e+00 11.741827087 -0.2700559 7.878041e-01
TypeVan            -1.878289e+01 27.646402307 -0.6793971 4.988232e-01
Weight:TypeLarge    2.419781e-03  0.005036981  0.4804029 6.322358e-01
Weight:TypeMidsize  7.724377e-04  0.004011302  0.1925653 8.477814e-01
Weight:TypeSmall   -6.858783e-03  0.004257682 -1.6109196 1.110875e-01
Weight:TypeSporty   6.787099e-04  0.004013261  0.1691168 8.661264e-01
Weight:TypeVan      4.283285e-03  0.007555762  0.5668898 5.723567e-01
```

And with Van as the reference level of type:

```r
mod_car93_full2<-lm(MPG.highway~Weight*Type2, data=Cars93)
summary(mod_car93_full2)$coef
```

```
                       Estimate   Std. Error    t value  Pr(>|t|)
(Intercept)         29.035823951 25.696130443  1.1299687 0.2618244
Weight              -0.001865770  0.006703380 -0.2783327 0.7814662
Type2Compact        18.782885727 27.646402307  0.6793971 0.4988232
Type2Large          11.472811300 29.009700360  0.3954819 0.6935270
Type2Midsize        15.971945860 26.574099715  0.6010343 0.5494954
Type2Small          36.525636382 26.318288216  1.3878424 0.1689910
Type2Sporty         15.611936118 26.346331477  0.5925658 0.5551224
Weight:Type2Compact -0.004283285  0.007555762 -0.5668898 0.5723567
Weight:Type2Large   -0.001863504  0.007625761 -0.2443696 0.8075625
Weight:Type2Midsize -0.003510847  0.006990822 -0.5022080 0.6168838
Weight:Type2Small   -0.011142068  0.007135048 -1.5615968 0.1222818
Weight:Type2Sporty  -0.003604575  0.006991947 -0.5155324 0.6075853
```

On Your Own

1. Continuing with the examples above in Section :

    a. What is the linear function to predict the highway fuel efficiency of a large car that weighs 3800 lbs using the model with Compact cars as the reference level?

    b. Confirm the linear function from (a) still applies if you instead use the model fit with Vans as the reference level.

    c. If the reference level is changed to Sporty, what do you expect the `TypeSmall` and `Weight:TypeSmall` coefficient estimates to be? Explain the reasoning and answer without running the new model.

    d. Create a plot showing the six fit lines predicting highway mpg as a function of vehicle weight - one line per vehicle type. You'll need the `Cars93` data set contained in the `MASS` library.

    e. Are all assumptions necessary for inference met? Explain.

2. Consider all cats with a body weight 3.5kg or less in the `cats` data frame from the MASS library.

    a. Fit a simple linear regression model estimating a cat's heart weight as a function of body weight. What is the equation of your fitted model?

    b. Are all assumptions for inference met by your model in (a)? Explain.

c. Is there sufficient evidence that the slope is greater than 3? Explain.

d. Now fit a model estimating heart weight using body weight interacting with cat sex. What is the equation of the fitted model?

e. Are all assumptions for inference met by your model in (d)? Explain.

f. Is there sufficient evidence that the slope is greater than 3 for male cats? Is there evidence that the slope is less than 3 for female cats? Use an $\alpha = 0.05$ level and explain your findings thoroughly.

g. Create a plot showing both the male and female cat fit lines. Jitter your body weights and use either plotting characters or color to distinguish which sex applies to each data point.

h. Why is it perfectly reasonable to calculate a prediction interval for the heart weight of a male cat weighing 3.3 kg, but not a good idea to make a prediction interval for a female cat weighing 3.3 kg?

3. The crabs data set in the MASS library contains a variety of body measurements on two species of crabs.

   a. Estimate a crab's body depth as a function of carapace width, including crab species information in your model. What type of model makes the most sense: a model with an additive inclusion of a species indicator, or a model with a multiplicative interaction with a species indicator? Explain your answer.

   b. What is the fitted equation of your selected model?

   c. Do Orange crabs and Blue crabs have a significantly different intercept in a linear model using carapace width to explain body depth? Explain.

   d. Are all assumptions for inference met in your model? Explain, using an $\alpha = 0.01$ significance level in any tests you run. If your model needs fixing for inference procedures to be valid, fix it.

   e. Using a model that meets all assumptions at the $\alpha = 0.01$ level, fit and interpret a 90% confidence interval for the mean body depth of a Blue crab with a carapace width of 40mm. Then fit and interpret a 90% confidence interval for the mean body depth of an Orange crab with a carapace width of 40mm. How do they compare?

   f. Now complete a model predicting body depth as a function of carapace width using an indicator term for crab sex. What indicator term is more helpful in predicting body depth: sex or species? Explain.

4. The `openintro` library includes a data frame called fastfood that contains nutrition information for 515 fast food items.

a. Fit a model for food item calories as a function of total fat content. Include restaurant as an interaction term in your model. Which two restaurants have the most similar fit lines? What are those two fit lines?

b. Which restaurant has the steepest slope? Which has the least-steep slope? Are the two slopes significantly different from each other?

c. Fit a new model that does not include restaurant. Which restaurant from your first model comes closest to matching this restaurant-blind overall fit for calories as a function of fat? Explain.

d. Create a plot showing the raw data with three lines: the fit for the restaurant with the steepest slope, the fit for the restaurant with the least-steep slope, and the fit for the restaurant that most closely matched the restaurant neutral fit. Include a legend that provides the name of the restaurant associated with each model fit.