Group-based models in R

Dave Dubin

September, 2018

Section 4.6.1 of the Kaplan text explains how to create a group-based model using the **mm** command, but unfortunately **mm** is no longer supported in the mosaic package. As we discussed in class, once we've computed the means for the groups we're using as the basis for our model, we can reach the same results shown in section 4.6.1 by adding two new columns to our data frame, one for the fitted model values, and one for the residuals.

Following along with the Kaplan text, the first thing to do is to load the mosaic package and read our data into a data frame. As discussed in class, we'll use the **read.csv** command instead of **fetchData**.

```
> require(mosaic)
> kids = read.csv("kidsfeet.csv")
```

We'll next compute the group means using mosaic's **mean** command instead of **mm** and confirm that the resulting data structure indexes the means by the levels of the grouping variable.

```
> km = mean(kids$width ~ kids$sex)
> km
       B        G
9.190000 8.784211
```

Knowing this, we can now interpret **km** as a discrete function from a domain of **sex** values to a codomain of means. But with these group-based models the means *are* the model values we want to copy into a new **fitted** column in the data frame. So first we wrap **km** in an R scalar function definition and confirm that we can retrieve the mean width for girls.

```
> kmf <- function(x){return(km[[x]])}
> kmf("G")
[1] 8.784211
```

Although it is possible to write loops in R, the best way to apply this function over an entire column is to use R's **sapply** command. We can now define a vector function, and confirm that it maps from the **sex** column of the data frame to a vector of means.

```
> kmff <- function(v){sapply(v,kmf)}
> kids$sex
 [1] B B B B B B B G G B B B B B G G G G G G B B G G G B G B B B G G G B B G
G G
[39] G
Levels: B G
> kmff(kids$sex)
```

```
 [1] 9.190000 9.190000 9.190000 9.190000 9.190000 9.190000 9.190000 8.784211
 [9] 8.784211 9.190000 9.190000 9.190000 9.190000 9.190000 8.784211 8.784211
[17] 8.784211 8.784211 8.784211 8.784211 9.190000 9.190000 8.784211 8.784211
[25] 8.784211 9.190000 8.784211 9.190000 9.190000 9.190000 8.784211 8.784211
[33] 8.784211 9.190000 9.190000 8.784211 8.784211 8.784211 8.784211
```

Now we're ready to compute the variance of fitted values and residuals as described on page 84 of Kaplan. As in previous Chapter 4 examples, we'll use the **transform** command to add **fitted** and **residual** columns to the **kids** data frame.

```
> kids = transform(kids,fitted=kmff(sex))
> kids = transform(kids,resid=(width - fitted))
```

You can view the data to confirm they're there. Next we follow Kaplan's example demonstrating the *partioning property* for variance with respect to these group-based models. The variance of the model values and the variance of the residuals sum to the variance of the width variable. We see that grouping by sex accounts for just over sixteen percent of the variance of feet width.

```
> var(kids$fitted)
[1] 0.04222182
> var(kids$resid)
[1] 0.2174543
> var(kids$fitted) + var(kids$resid)
[1] 0.2596761
> var(kids$width)
[1] 0.2596761
> (var(kids$fitted))/(var(kids$width))
[1] 0.1625942
```