Interaction Terms in Regression

Mike Levine

February 10, 2019

Note: *this R code was written as an assignment for Advanced Business Statistics (STAT 5607-308) at Temple University's Master of Science in Business Analytics program.*

Interaction terms are commonly used in regression to better model the relationship between two independent variables. In regression, interaction terms are used when the effect of the dependent variable on one independent variable changes based on the value of another independent variable. For example, let's suppose we're trying to predict profit for a company; two of our independent variables are "Seasonal Sales" and "Month." Seasonal sales for a company may be higher in the summer and lower in the winter. Using an interaction term in a regression model helps us to better illustrate this relationship.

Now let's suppose we're trying to predict the future college GPA of high school students. In the data below, we use the mock 'GPA2' dataset containing 4,135 college students. The variables in the dataset are as follows:

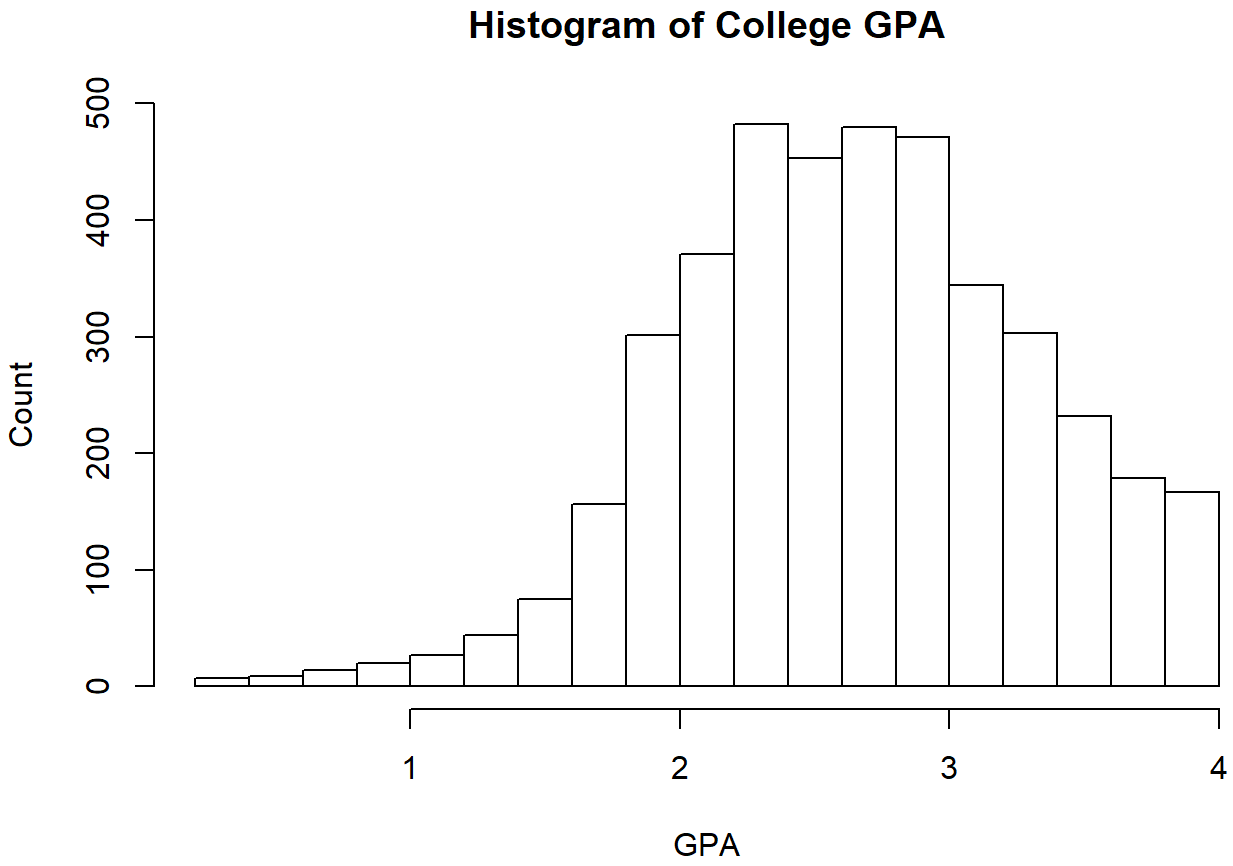
- **colgpa**: college GPA
- **SAT**: combined SAT score
- **athlete**: =1 if athlete
- **hsize**: size of graduating high school class
- **hsrank**: rank in high school graduating class
- **hsperc**: high school percentile, from the top
- **female**: =1 if female

First, let's set our working directory. We'll also read our dataset into R and call it "dat."

```
setwd("C:/Users/mdlev/OneDrive/Documents/Education/Graduate - Temple
University/2nd Semester/Advanced Business Statistics/R Datasets")
getwd()
## [1] "C:/Users/mdlev/OneDrive/Documents/Education/Graduate - Temple
University/2nd Semester/Advanced Business Statistics/R Datasets"
dat <- read.csv("GPA2.csv")
```

Next, let's create a histogram of our 'colgpa' variable to make sure the data is normally distributed. While the variable is skewed to the left, it isn't dramatic enough to warrant a transformation (log, square root, etcetera). Therefore, we'll leave it as-is.

```
hist(dat$colgpa, main = "Histogram of College GPA", xlab = "GPA", ylab =
"Count")
```



Next, let's create a correlation matrix of our 'GPA2' dataset. From this, we can see that 'Hsize' and 'hsrank,' as well as 'hsperc' and 'hsrank,' both have particularly high correlations (0.61).

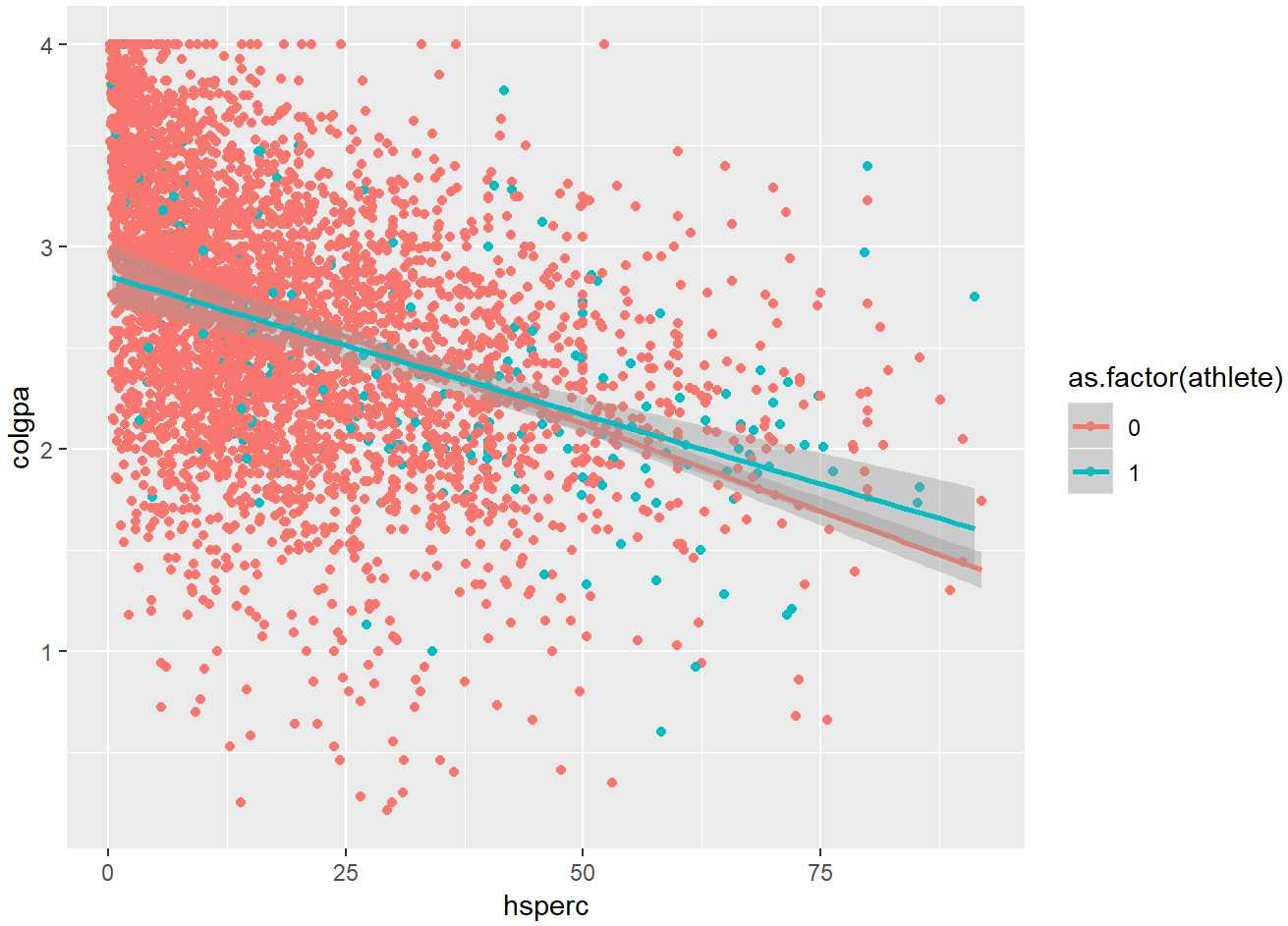
```
round(cor(dat),2)
##         colgpa   SAT athlete hsize hsrank hsperc female
## colgpa    1.00  0.41   -0.09 -0.03  -0.33  -0.43   0.11
## SAT       0.41  1.00   -0.19  0.06  -0.18  -0.28  -0.15
## athlete  -0.09 -0.19    1.00  0.05   0.19   0.20  -0.10
## hsize    -0.03  0.06    0.05  1.00   0.61  -0.04   0.00
## hsrank   -0.33 -0.18    0.19  0.61   1.00   0.61  -0.10
## hsperc   -0.43 -0.28    0.20 -0.04   0.61   1.00  -0.15
## female    0.11 -0.15   -0.10  0.00  -0.10  -0.15   1.00
```

Next, we'll run an initial regression to predict college GPA (y) using all of the predictor variables in our dataset. SAT, athlete, hsrank, hsperc and female seem important in the model. Their p-values are less than 0.05, and their coefficients show significant changes (both negative and positive). However, our adjusted R-squared is 0.2959. This means our regression model is explaining 29.59% of the variation in our data. Thus, our model needs to be strengthened for it to be managerially relevant.

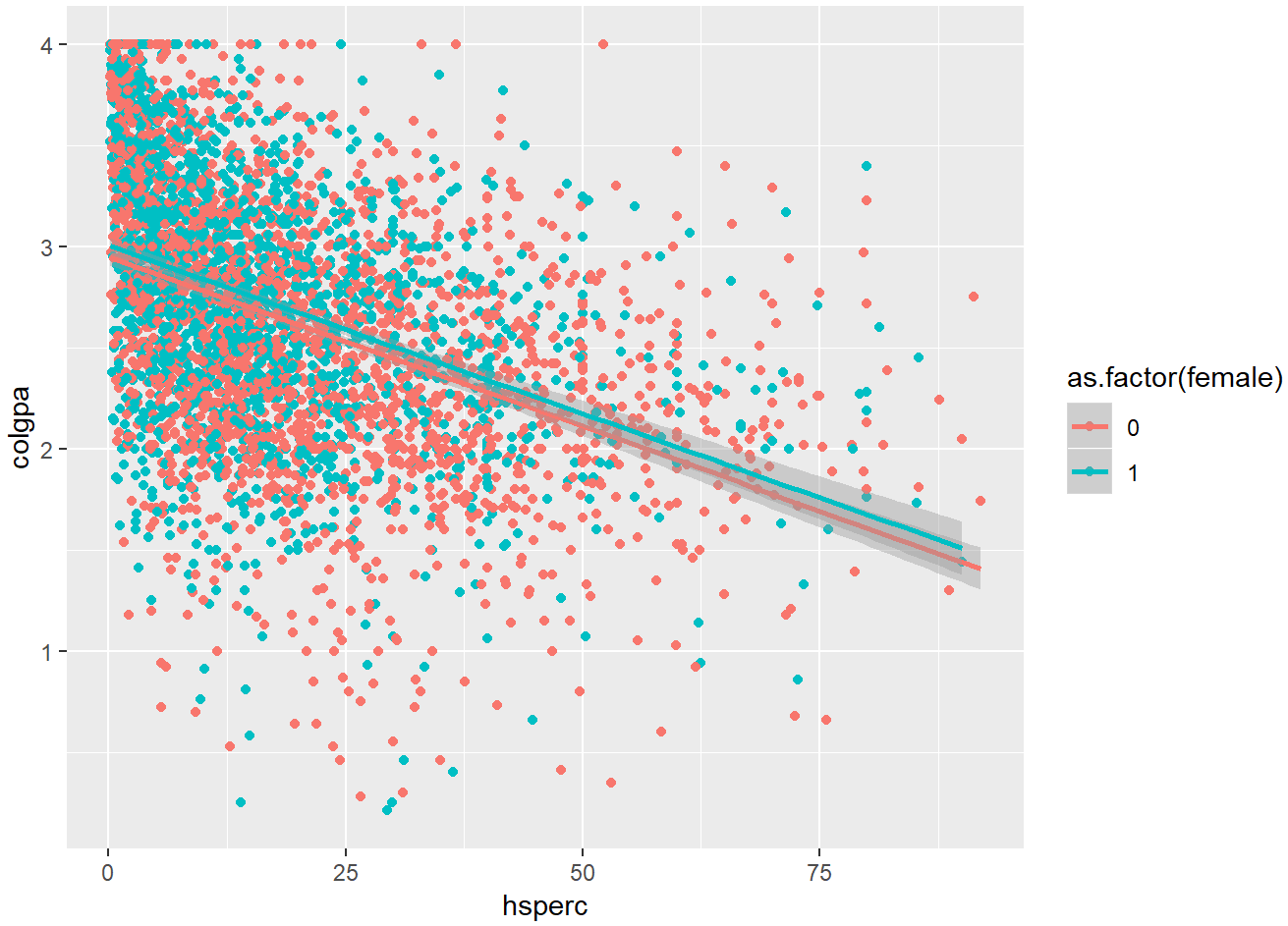
```
reg1 <- lm(colgpa ~ SAT + athlete + hsize + hsrank + hsperc + female, data =
dat)
summary(reg1)
##
## Call:
## lm(formula = colgpa ~ SAT + athlete + hsize + hsrank + hsperc +
##     female, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69533 -0.35025  0.02885  0.38460  1.91527
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.136e+00  7.643e-02  14.870  < 2e-16 ***
## SAT          1.621e-03  6.659e-05  24.346  < 2e-16 ***
## athlete      1.767e-01  4.210e-02   4.197 2.76e-05 ***
## hsize        9.934e-05  8.211e-05   1.210    0.226
## hsrank      -1.436e-03  2.785e-04  -5.155 2.65e-07 ***
## hsperc      -9.434e-03  8.698e-04 -10.846  < 2e-16 ***
## female       1.514e-01  1.788e-02   8.469  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5506 on 4128 degrees of freedom
## Multiple R-squared:  0.2969, Adjusted R-squared:  0.2959
## F-statistic: 290.5 on 6 and 4128 DF,  p-value: < 2.2e-16
```

Let's look at some possible interaction terms to strengthen our model. The interaction between 'hsperc' and 'colgpa' broken down by 'athlete' is worth exploring since the two lines intersect. The second scatterplot, broken down by 'female,' doesn't appear to have intersecting lines and therefore probably isn't worth exploring.

```
library(ggplot2)
ggplot(dat, aes(x = hsperc, y = colgpa, color = as.factor(athlete))) +
   geom_point() + geom_smooth(method = "lm")
```




```
ggplot(dat, aes(x = hsperc, y = colgpa, color = as.factor(female))) +
   geom_point() + geom_smooth(method = "lm")
```



Now let's create a revised model with 'hsperc' and 'athlete' as interaction terms. Being an athlete, when considered as an interaction term on 'hsperc', increases colgpa by 0.006 for every increase of 1 percentile.

```
reg2 <- lm(colgpa ~ SAT + hsrank + female + hsperc * athlete, data = dat)
summary(reg2)
##
## Call:
## lm(formula = colgpa ~ SAT + hsrank + female + hsperc * athlete,
##     data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69341 -0.34768  0.03148  0.38268  1.92715
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.1557274  0.0750077  15.408  < 2e-16 ***
## SAT             0.0016370  0.0000662  24.728  < 2e-16 ***
## hsrank         -0.0012024  0.0001683  -7.146 1.06e-12 ***
## female          0.1521870  0.0178538   8.524  < 2e-16 ***
## hsperc         -0.0105123  0.0006918 -15.195  < 2e-16 ***
## athlete        -0.0204410  0.0748272  -0.273  0.78473
```

```
## hsperc:athlete  0.0059980  0.0018802   3.190  0.00143 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5501 on 4128 degrees of freedom
## Multiple R-squared:  0.2984, Adjusted R-squared:  0.2974
## F-statistic: 292.6 on 6 and 4128 DF,  p-value: < 2.2e-16
```

According to our revised model, if 'female' increases by 1 (meaning if the datapoint is for a female college student, since the variable is binary) then 'colgpa' increases by 0.1522. Likewise, if 'SAT' increases by 1 then 'colgpa' increases by 0.0016. Our least squares line can be written as: colgpa = 1.1557(SAT) + 0.0016(hsrank) + 0.1522(female) + -0.0105(hsperc) * (athlete).

Now let's suppose a male student received an 1120 on his SAT and was ranked 24th in his high school class, which was the 12th percentile. Let's also assume he was an athlete. Using our model, our predicted college GPA for this student is 2.886. The lower threshold is 1.8014 and the upper threshold is 3.9699.

```
newdat <- data.frame(SAT = 1120, hsrank=24, hsperc = 12, athlete = 1,
female=0)
predict(reg2, newdat, interval = "predict")
##        fit      lwr      upr
## 1 2.885685 1.801442 3.969929
```

Overall, our model attempts to demonstrate that SAT scores, high school rank, gender, high school percentile, and whether or not the student is an athlete each act as predictors for a student's college GPA.

While our model accounts for a number of interesting relationships, it is not managerially useful in its present form. The adjusted R-squared value is 0.2974. This means that only 30% of the variation in the data is accounted for by our variables. That leaves roughly 70% of the variation unaccounted for.