Chapter 6

Multiple Regression Analysis - Further Issues

Jim Bang

Interactions between Variables

Interactions between Continous Variables

Suppose we want to estimate the following model of (standardized) final exam scores:

$$Final = \beta_0 + \beta_1 Attendance + \beta_2 PriorGPA + \beta_3 ACT + \beta_4 PriorGPA^2 + \beta_5 ACT^2$$

where:

$Attendance$ = percentage of classes the student attended, `atndrte` $PriorGPA$ = student's GPA in previous classes, `priGPA` ACT = student's ACT score, `ACT`

We may suspect that the *effect* of attendance depends on, or is *moderated by*, prior GPA: Attendance might not matter as much for students who do perform well generally. Then,

$$Final = \beta_0 + \beta_1 Attendance + \beta_2 PriorGPA + \beta_3 ACT + \beta_4 PriorGPA^2 + \beta_5 ACT^2 + \beta_6 PriorGPA \cdot Attendance + u$$

$\hat{\beta}_1$ = effect of attendance on final performance *when prior GPA equals zero.*

What we (probably) want to know: The effect of attendance on final performance *at the mean* (of prior GPA and other variables): $\frac{\delta Final}{\delta Attendance}\Big|_{PriorGPA=\bar{x}_{PriorGPA}} = \beta_1 + \beta_6 \cdot \bar{x}_{PriorGPA}$

Interactions in `R`

1. Estimate the above regressions using the (pre-loaded) `attend` dataset (`attend.lm1` and `attend.lm2`).
2. Summarize the results in a `stargazer` text table.
3. Calculate $\frac{\delta Final}{\delta Attendance}\Big|_{PriorGPA=\bar{x}_{PriorGPA}}$ for each model.

```r
attend.lm1 <- lm(stndfnl ~ atndrte + priGPA + ACT + I(priGPA^2) + I(ACT^2), data = attend)
attend.lm2 <- lm(stndfnl ~ atndrte*priGPA + ACT + I(priGPA^2) + I(ACT^2), data = attend)
attend.lm1$coefficients['atndrte']
```

```
##     atndrte
## 0.006174715
```

```r
attend.lm2$coefficients['atndrte'] + attend.lm2$coefficients['atndrte:priGPA'] * mean(attend.lm1$model$priGPA)
```

```
##     atndrte
## 0.007736558
```

Centering Continous Interactions

A trick to make things less cumbersome is to *center* the variables in the interaction. This estimates:

$$Final = \beta_0 + \beta_1 Attendance + \beta_2 PriorGPA + \beta_3 ACT + \beta_4 PriorGPA^2 + \beta_5 ACT^2 + \beta_6 (PriorGPA - \mu_{PriorGPA}) \cdot (Attendance - \mu_{Attendance}) + u$$

Re-estimate `attend.lm2` using the `scale()` function with `scale = FALSE` and call it `attend.lm3`.

Summarize all three *attend* models using the stargazer command.

```r
attend.lm3 <- lm(stndfnl ~ atndrte + priGPA + ACT + I(priGPA^2) + I(ACT^2) + I(scale(atndrte, scale = FALSE)*scale(priGPA, scale = FALSE))
stargazer(attend.lm1, attend.lm2, attend.lm3, type = 'text')
```

```
##
## =======================================================================================================================================
##                                                                                            Dependent variable:
##                                                                  ----------------------------------------------------------------------
##                                                                                                  stndfnl
##                                                                            (1)                     (2)                     (3)
## ---------------------------------------------------------------------------------------------------------------------------------------
## atndrte                                                                 0.006***                 -0.007                 0.008***
##                                                                          (0.002)                 (0.010)                 (0.003)
##
## priGPA                                                                  -1.491***               -1.629***               -1.172**
##                                                                          (0.469)                 (0.481)                 (0.530)
##
## ACT                                                                      -0.118                  -0.128                  -0.128
##                                                                          (0.098)                 (0.098)                 (0.098)
##
```

```
## I(priGPA2)                                                             0.359***                0.296***                0.296***
##                                                                         (0.089)                 (0.101)                 (0.101)
##
## I(ACT2)                                                                 0.004**                 0.005**                 0.005**
##                                                                         (0.002)                 (0.002)                 (0.002)
##
## atndrte:priGPA                                                                                   0.006
##                                                                                                 (0.004)
##
## I(scale(atndrte, scale = FALSE) * scale(priGPA, scale = FALSE))                                                          0.006
##                                                                                                                         (0.004)
##
## Constant                                                                 1.296                   2.050                   0.870
##                                                                         (1.230)                 (1.360)                 (1.273)
##
## -------------------------------------------------------------------------------------------------------------------------------------------
## Observations                                                              680                     680                     680
## R2                                                                       0.227                   0.229                   0.229
## Adjusted R2                                                              0.221                   0.222                   0.222
## Residual Std. Error                                                0.873 (df = 674)        0.873 (df = 673)        0.873 (df = 673)
## F Statistic                                                     39.526*** (df = 5; 674) 33.250*** (df = 6; 673) 33.250*** (df = 6; 673)
## ===========================================================================================================================================
## Note:                                                                                                           *p<0.1; **p<0.05; ***p<0.01
```

Comparing Specifications

Adjusted R^2

$$\bar{R}^2 = 1 - \frac{SSR/(n-k-1)}{SST/(n-1)} = 1 - \frac{\hat{\sigma}_u^2}{\hat{\sigma}_y^2}$$

Useful for comparing nonnested models, e.g. switching one control for another, and alternate forms of the dependent variable.

Information Criteria for Nested Models

Choose the *Lowest*-Valued Specification

Akaike Information Criteria (Normal Error Distribution): $-2\frac{\ell}{n} + 2\frac{k}{n} = C + ln(\frac{\sum_{i=1}^{n} u_i^2}{n}) + 2\frac{k}{n}$

Bayes/Schwartz Information Criteria (Normal Error Distribution): $-2\frac{\ell}{n} + ln(n)\frac{k}{n} = C + ln(\frac{\sum_{i=1}^{n} u_i^2}{n}) + ln(n)\frac{k}{n}$

Inappropriate Controls

Adding a control that directly causes y and is *uncorrelated* with x generally helps by reducing $\hat{\sigma}_u^2$.

Adding a control that directly causes y and is *correlated* with x is tricky: 1. Adding the control helps by reducing omitted-variable bias; 2. Adding the control hurts by increasing the variance.

Things get even trickier when the control only *indirectly* causes y through x (collider bias), or when x inflences the control on the way to causing the outcome (post-treatment effect bias).

Be careful adding controls!

Overfitting

Controlling for too many things risks overfitting the sample (and threatens exteernal validity).

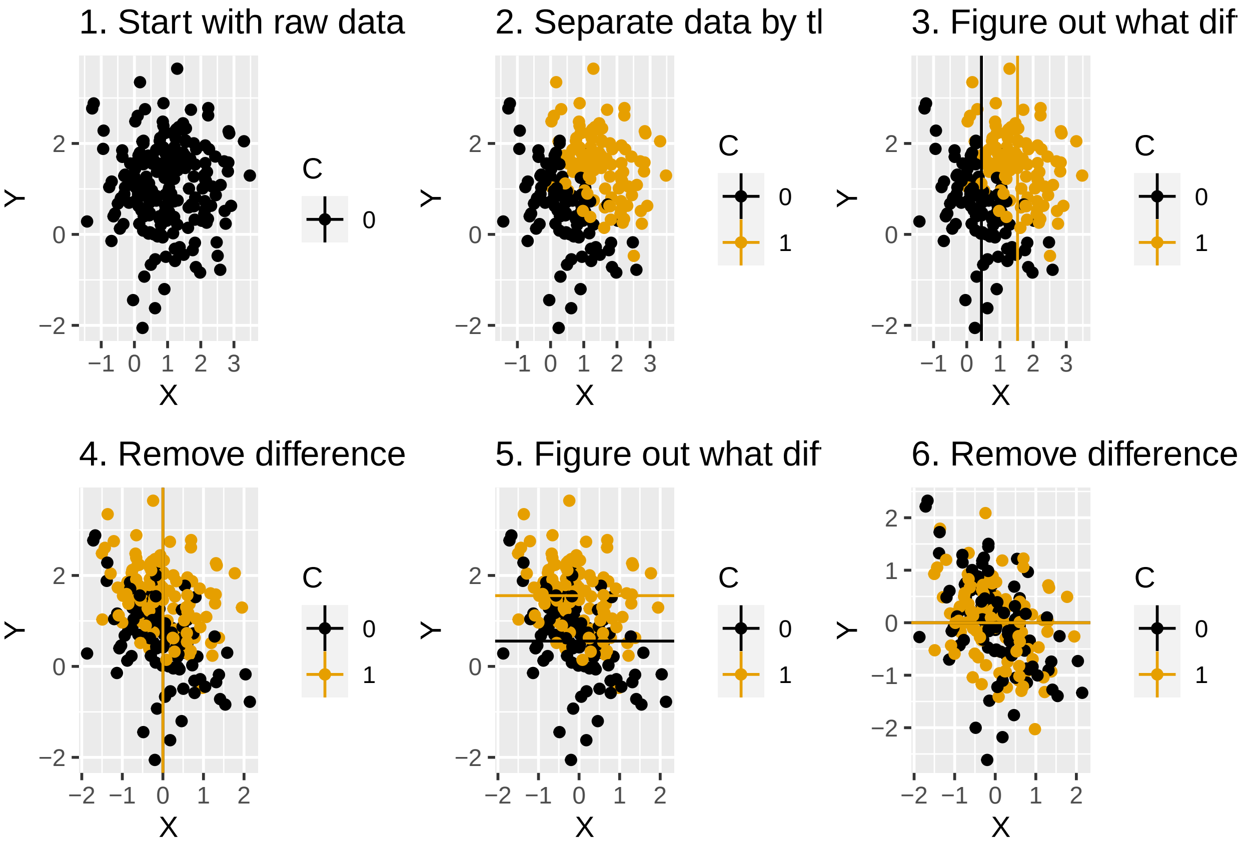
1. Forward/Backward Stepwise Regression
2. Ridge Regression
3. Least-Angle Regression (LARS)
4. Least Absolute Shrinkage & Selection Operator (LASSO)
5. Validation Sample/Cross-Validation

Colliders

Controlling for a variable that *causes* your treatment introduces *collider bias*.

```r
df <- data.frame(X = rnorm(200)+1,Y=rnorm(200)+1,time="1") %>%
  mutate(C = as.integer(X+Y+rnorm(200)/2>2)) %>%
  group_by(C) %>%
  mutate(mean_X=mean(X),mean_Y=mean(Y)) %>%
  ungroup()
before_cor <- paste("1. Start with raw data, ignoring C. Correlation between X and Y: ",round(cor(df$X,df$Y),3),sep='')
after_cor <- paste("7. Analyze what's left! Correlation between X and Y controlling for C: ",round(cor(df$X-df$mean_X,df$Y-df$mean_Y),3),s
dffull <- rbind(
  df %>% mutate(mean_X=NA,mean_Y=NA,C=0,time=before_cor),
  df %>% mutate(mean_X=NA,mean_Y=NA,time='2. Separate data by the values of C.'),
  df %>% mutate(mean_Y=NA,time='3. Figure out what differences in X are explained by C'),
  df %>% mutate(X = X - mean_X,mean_X=0,mean_Y=NA,time="4. Remove differences in X explained by C"),
  df %>% mutate(X = X - mean_X,mean_X=NA,time="5. Figure out what differences in Y are explained by C"),
  df %>% mutate(X = X - mean_X,Y = Y - mean_Y,mean_X=NA,mean_Y=0,time="6. Remove differences in Y explained by C"),
  df %>% mutate(X = X - mean_X,Y = Y - mean_Y,mean_X=NA,mean_Y=NA,time=after_cor))
p1 <- ggplot(subset(dffull, time == names(table(dffull$time))[1]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[1])
p2 <- ggplot(subset(dffull, time == names(table(dffull$time))[2]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[2])
p3 <- ggplot(subset(dffull, time == names(table(dffull$time))[3]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[3])
```

```
p4 <- ggplot(subset(dffull, time == names(table(dffull$time))[4]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[4])
p5 <- ggplot(subset(dffull, time == names(table(dffull$time))[5]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[5])
p6 <- ggplot(subset(dffull, time == names(table(dffull$time))[6]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[6])
ggarrange(p1, p2, p3, p4, p5, p6, nrow = 2, ncol = 3)
```

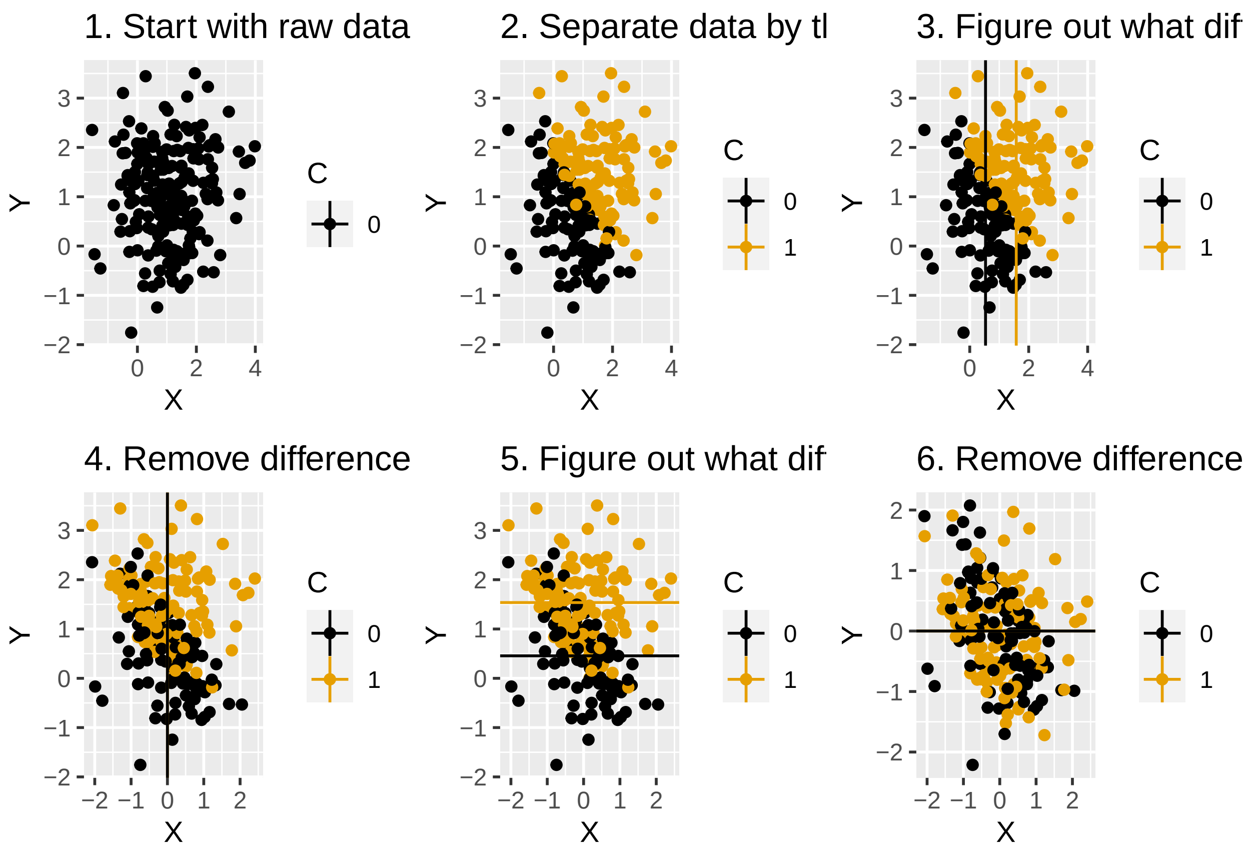

1. Start with raw data
2. Separate data by tl
3. Figure out what dif
4. Remove difference
5. Figure out what dif
6. Remove difference
Y
X
C
0
1
2
−2
−1
3

Post-Treatment Effects

Controlling for a variable that is *caused by* the treatment and subsequently causes the outcome introduces post-treatment effect bias.

```
df <- data.frame(X = rnorm(200)+1,Y=rnorm(200)+1,time="1") %>%
  mutate(C = as.integer(X+Y+rnorm(200)/2>2)) %>%
  group_by(C) %>%
  mutate(mean_X=mean(X),mean_Y=mean(Y)) %>%
  ungroup()
before_cor <- paste("1. Start with raw data, ignoring C. Correlation between X and Y: ",round(cor(df$X,df$Y),3),sep='')
after_cor <- paste("7. Analyze what's left! Correlation between X and Y controlling for C: ",round(cor(df$X-df$mean_X,df$Y-df$mean_Y),3),s
dffull <- rbind(
  df %>% mutate(mean_X=NA,mean_Y=NA,C=0,time=before_cor),
  df %>% mutate(mean_X=NA,mean_Y=NA,time='2. Separate data by the values of C.'),
  df %>% mutate(mean_Y=NA,time='3. Figure out what differences in X are explained by C'),
  df %>% mutate(X = X - mean_X,mean_X=0,mean_Y=NA,time="4. Remove differences in X explained by C"),
  df %>% mutate(X = X - mean_X,mean_X=NA,time="5. Figure out what differences in Y are explained by C"),
  df %>% mutate(X = X - mean_X,Y = Y - mean_Y,mean_X=NA,mean_Y=0,time="6. Remove differences in Y explained by C"),
  df %>% mutate(X = X - mean_X,Y = Y - mean_Y,mean_X=NA,mean_Y=NA,time=after_cor))
p1 <- ggplot(subset(dffull, time == names(table(dffull$time))[1]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[1])
p2 <- ggplot(subset(dffull, time == names(table(dffull$time))[2]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[2])
p3 <- ggplot(subset(dffull, time == names(table(dffull$time))[3]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[3])
```

```r
p4 <- ggplot(subset(dffull, time == names(table(dffull$time))[4]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[4])
p5 <- ggplot(subset(dffull, time == names(table(dffull$time))[5]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[5])
p6 <- ggplot(subset(dffull, time == names(table(dffull$time))[6]),
  aes(y=Y,x=X,color=as.factor(C)))+geom_point()+
  geom_vline(aes(xintercept=mean_X,color=as.factor(C)), na.rm = TRUE)+
  geom_hline(aes(yintercept=mean_Y,color=as.factor(C)), na.rm = TRUE)+
  guides(color=guide_legend(title="C"))+
  scale_color_colorblind()+
  labs(title = names(table(dffull$time))[6])
ggarrange(p1, p2, p3, p4, p5, p6, nrow = 2, ncol = 3)
```



Further Issues

- Prediction Intervals
- Residual Analysis
- Predicting y when $log(y)$ is the dependent variable