HW3 Answers

Alan Spearot

3/2/2019

Problem 1

Part A

First, we load and clean our data:

```
library(foreign)
data_org<-read.dta("https://people.ucsc.edu/~aspearot/Econ_217/org_example.dta")
data_sub<-subset(data_org, (state=="CA"|state=="NV")&(year==2008|year==2013))
data_sub<-subset(data_sub, rw>0 & is.na(educ)==FALSE&is.na(age)==FALSE)
summary(data_sub)
```

Then, we define the dummy variables to run the diff and diff:

```
ca<-as.numeric(data_sub$state=="CA")
yr13<-as.numeric(data_sub$year=="2013")
```

Now we generate the interaction terms. There are a number of ways to do this in R.

1. Generate a new variable, say, interaction=ca*yr13, then add it into regression equation
2. Use ":" ca:yr13
3. Use "*", it will be automatically converted to":"
4. Use I()

We will use the latter, since we can then write the code within the glm command.

```
DID_glm<-lm(log(rw)~ca+yr13+I(ca*yr13)+educ+age,data_sub)
summary(DID_glm)
```

```
##
## Call:
## lm(formula = log(rw) ~ ca + yr13 + I(ca * yr13) + educ + age,
##     data = data_sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.07560 -0.34780 -0.02833  0.33719  2.48565
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.0106489  0.0224299  89.641  < 2e-16 ***
## ca                0.0347036  0.0158356   2.191   0.0284 *
## yr13             -0.0931132  0.0212147  -4.389 1.15e-05 ***
## I(ca * yr13)      0.0368266  0.0230641   1.597   0.1104
```

```
## educHS            0.2926381  0.0147493  19.841  < 2e-16 ***
## educSome college  0.4257850  0.0143603  29.650  < 2e-16 ***
## educCollege       0.8080216  0.0150892  53.550  < 2e-16 ***
## educAdvanced      1.0765064  0.0175361  61.388  < 2e-16 ***
## age               0.0102059  0.0003098  32.943  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5316 on 16294 degrees of freedom
## Multiple R-squared:  0.3151, Adjusted R-squared:  0.3147
## F-statistic:   937 on 8 and 16294 DF,  p-value: < 2.2e-16
```

The estimate suggests that California recovered roughly 3.7% faster than Nevada between the period 08 and 13. Printing the confidence interval:

```r
print(confint(DID_glm,4,level=.95))
```

```
##                      2.5 %     97.5 %
## I(ca * yr13) -0.008381561 0.08203467
```

Part B

First, create a function to randomly sample rows from a data frame. Don't forget to sample with replacement!

```r
randomSample = function(df,n) {
  return (df[sample(nrow(df),n, replace=TRUE),])
}
```

Next, run the bootstrap resampling, as discussed in the lecture notes.

```r
rm(results)
for(rep in 1:1000){
  fit.B<-lm(log(rw)~ca+yr13+I(ca*yr13)+educ+age,randomSample(data_sub,nrow(data_sub)))
  coef.B<-as.numeric(coef(fit.B))
  save.B<-data.frame(rep,t(as.matrix(coef.B)))
  if(rep%%10==0){print(rep)}
  if(rep==1){results<-save.B}
  if(rep>1){results<-rbind(results,save.B)}
}
```

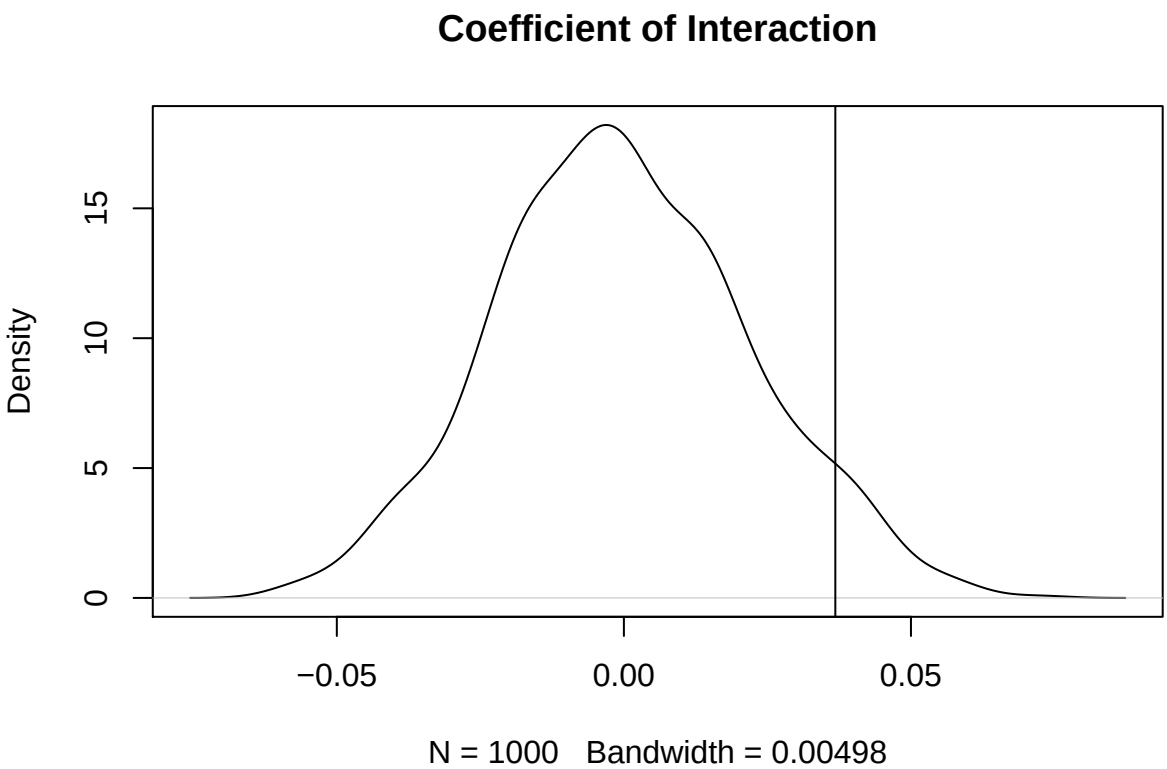
Next, construct a 95% confidence interval:

```r
quantile(results[,5],prob=c(.025,.975),na.rm=TRUE)
```

```
##        2.5%       97.5%
## -0.04214743  0.04406807
```

And then plot the distribution to have a look at how the estimates have changed via the resampling procedure.

```r
plot(density(results[,5]),main="Coefficient of Interaction")
abline(v=coef(DID_glm)[4])
```



It appears that the confidence interval is approximately as wide but shifted downward by about 0.04, and cenetered around zero. So, the original effect isn't particularly robust to the bootstrap resampling procedure.

Part C

The key idea is, do a placebo test before the "treatment" window (2008-2013). You can use a number of different windows to get full marks, but for the answer key we will use 2003-2008. Creating the new subset of data:

```r
data_sub2<-subset(data_org, (state=="CA" | state=="NV") & (year==2003 | year==2008))
data_sub2<-subset(data_sub2, rw>0 & is.na(educ)==FALSE&  is.na(age)==FALSE)
```

Then, follow the same procedure, creating the dummy variables, then running the pre-trend regression:

```r
ca2<-as.numeric(data_sub2$state=="CA")
yr08<-as.numeric(data_sub2$year=="2008")

pre_trend<-lm(log(rw)~ca2+yr08+I(ca2*yr08)+educ+age,data_sub2)
summary(pre_trend)
```

```
##
## Call:
## lm(formula = log(rw) ~ ca2 + yr08 + I(ca2 * yr08) + educ + age,
##     data = data_sub2)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00064 -0.33121 -0.02475  0.32644  2.05893
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.0428543  0.0203795 100.241  < 2e-16 ***
## ca2                0.0864026  0.0140105   6.167 7.13e-10 ***
## yr08               0.0275469  0.0186628   1.476    0.140
## I(ca2 * yr08)     -0.0501549  0.0208200  -2.409    0.016 *
## educHS             0.2812305  0.0138994  20.233  < 2e-16 ***
## educSome college   0.4284990  0.0136430  31.408  < 2e-16 ***
## educCollege        0.7795389  0.0145659  53.518  < 2e-16 ***
## educAdvanced       1.0030881  0.0177848  56.401  < 2e-16 ***
## age                0.0090838  0.0003183  28.538  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5203 on 15953 degrees of freedom
## Multiple R-squared:  0.2922, Adjusted R-squared:  0.2918
## F-statistic: 823.1 on 8 and 15953 DF,  p-value: < 2.2e-16
```

In interpreting the pre-trend interaction, there sure looks to be one. But, let's bootstrap it anyway. First, we generate the residuals and fitted values:

```
  residuals.full<-as.numeric(pre_trend$residuals)
  predict.full<-as.numeric(pre_trend$fitted.values)
```

Then, run the residual bootstrap:

```
rm(results)
# residual resampling
  for(rep in 1:1000){
    rand.resid<-sample(residuals.full, nrow(data_sub2),replace=TRUE)
    data_sub2$rw_boot<-predict.full+rand.resid
    fit.B<-lm(rw_boot~ca2+yr08+I(ca2*yr08)+educ+age,data_sub2)
    coef.B<-as.numeric(coef(fit.B))
    save.B<-data.frame(rep,t(as.matrix(coef.B)))
    if(rep%%10==0){print(rep)}
    if(rep==1){results<-save.B}
    if(rep>1){results<-rbind(results,save.B)}
  }
```

Generating the confidence interval:

```
  quantile(results[,5],prob=c(.025,.975),na.rm=TRUE)
```

There is clearly a pre-trend. From the confidence interval, California real wages are growing between 0.6% and 9.1% less than Nevada wages.

Problem 2

Part A

First we will load the necessary libraries and datasets

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rpart)
library(foreign)

d<-read.dta("https://people.ucsc.edu/~aspearot/Econ_217/org_example.dta")
d<-subset(d,is.na(nilf)==FALSE&is.na(educ)==FALSE&is.na(age)==FALSE&is.na(female)==FALSE)

d$unem1<-ifelse(d$unem==1,"unemployed","employed")
d$nilf3<-ifelse(d$nilf==1,"Out of Labor Force",ifelse(d$empl==0,"Unemployed","Employed"))
```

Generate textual variables to measure the groups - it will be easier to interpret on the classification tree:

```
d$married<-ifelse(d$married==1,"married","not married")
d$citizen<-ifelse(d$citizen==1,"citizen","non citizen")
d$gender<-ifelse(d$female==1,"female","male")
```

Create the training and testing datasets.

```
subtrain<-subset(d,year==2013&state=="CA")
subtest<-subset(d,year==2013&state!="CA")
```

Run the k-nearest neighbors model, and generate predictions. Test the predictions using the testing dataset.

```
model.knn <- train(nilf3~educ+gender+wbho+married+ownchild+citizen,data=subtrain,method="knn")
print(model.knn)
```

```
## k-Nearest Neighbors
##
## 13738 samples
##     6 predictor
##     3 classes: 'Employed', 'Out of Labor Force', 'Unemployed'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 13738, 13738, 13738, 13738, 13738, 13738, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.6383012  0.2431943
##   7  0.6395647  0.2448219
```

```
##   9  0.6394542  0.2439007
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

```
outcomes.knn <- predict(model.knn,subtest)
subtest$outcomes.knn<-as.character(outcomes.knn)
sum(subtest$outcomes.knn==subtest$nilf3)/nrow(subtest)
```

```
## [1] 0.6512447
```

Run the classification tree, and generate predictions. Test the predictions using the testing dataset.

```
model.tree <- rpart(nilf3~educ+gender+wbho+married+ownchild+citizen,data = subtrain, method = "class")
outcomes.tree <- predict(model.tree,subtest,type='class')
subtest$outcomes.tree<-as.character(outcomes.tree)
sum(subtest$outcomes.tree==subtest$nilf3)/nrow(subtest)
```

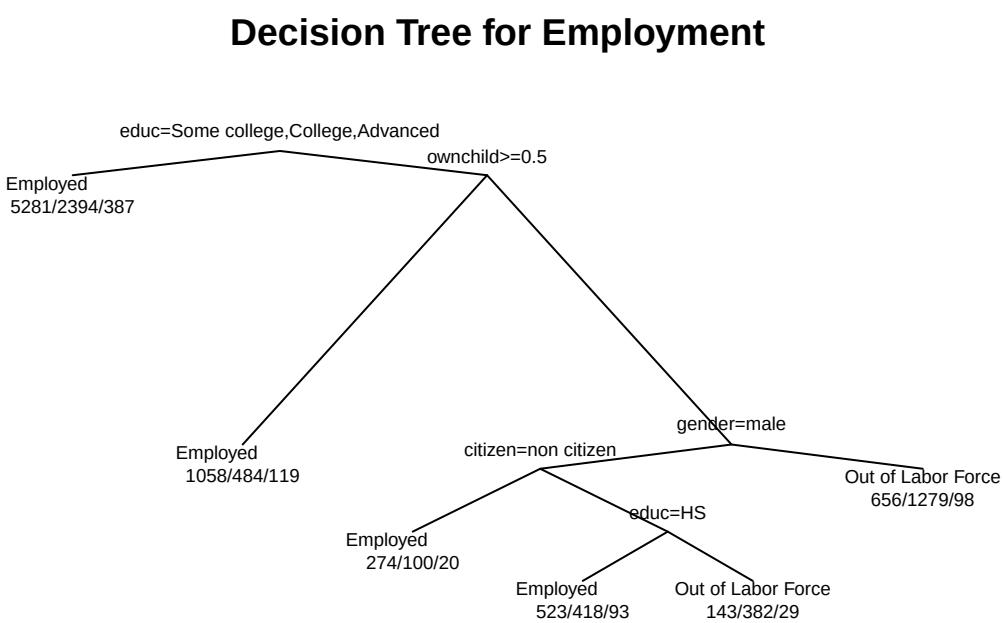
```
## [1] 0.654129
```

It appears that the classification tree does marginally better than the K-nearest neighbors model.

Part B

Generate the tree, with text labels.

```
plot(model.tree,cex=1,branch=0,main="Decision Tree for Employment",margin=.05)
text(model.tree,cex=0.6,use.n=TRUE,minlength=0)
```



To interpret the Figure, we first note that none of the predictions involve a respondent being unemployed! This is because the predominant two outcomes are being employed and being out of the labor force, and the model predictions focus on this.

Next, we see that having at least some college is a unifying predictor of being employed. If the respondent does not have at least some college, then more nuance is required. Conditional on having highschool or less, if you have at least one child then you are most likely to be employed. If not, then gender, citizenship status, and having a high school education matter for predicting between being employed and being out of the labor force.