Regression and Effective Samples

Drew Dimmery drewd@nyu.edu

February 13, 2015

Identification / Estimation

- The "Four questions" may have been a little confusing.
- Let me re-break down the last two questions as "Identification" and "Estimation"
- "A regression is causal when the CEF it approximates is causal" - MHE
- Identification consists (in this context) as the set of things you need to believe in order to believe the CEF is causal.
- Thus, when we think about identification, we should think about assumptions.
- Estimation is the process you use to estimate the CEF.

In practice

- The Conditional Independence Assumption would fall in identification.
- I'd also throw in assumptions of additivity / linearity in this pot.
- Estimation would include the populations/samples of interest, and all statistical inference.
- But there's a close connection between the two.
- Problems in estimation can lead to changes in identifying assumptions.
- So they aren't completely separable.

Covariate Adjustment in sampling

- Lin, Winston. (2013) "Agnostic Notes on Regression Adjustments to Experimental Data: Reexamining Freedman's Critique" *Annals of Applied Statistics*. 7(1):295-318.
- Imagine that we are biologists. We are interested in leaf size.

- Finding the size of leaves is hard, but weighing leaves is easy.
- Key insight is that we can use auxilliary information to be smarter:
  - Sample from leaves on a tree.
  - Measure their size and weight.
  - Let $\hat{y}_s$ be the average size in the sample.
  - Let $\hat{x}_s$ be the average weight in the sample.
  - Population averages drop the subscript.
  - We know that $\hat{y}_s$ unbiased and consistent for $\hat{y}$
  - But we have extra information!
  - We also have $\hat{x}$ (all the weights)
  - This motivates the regression estimator:
    $\hat{\hat{y}}_{reg} = \hat{y}_s + \beta(\hat{x} - \hat{x}_s)$
  - We get β by a regression of leaf area on weight in the sample.

Connection to Multiple Regression

- In the case of OLS for the analysis of experiments, we have nearly the same setup.
- Only difference is that we are sampling for both treatment and control.
- This means that we must adjust both groups separately.
- This motivates the use of covariate-treatment interactions.
- Remember! Freedman (2008) showed that regression is **biased** and can be **inconsistent** for an experimental parameter in the case when interactions aren't included.
- This is something to consider in many different situations. There's no reason to expect treatment and control groups to exhibit identical effects (even ones that are orthogonal to the causal parameter of interest)
- This is just a particular sort of omitted variable bias, which you should already be familiar with.

Covariate Adjustment in Experiments

- Now imagine we are social scientists (hopefully this isn't hard)
- We are interested in the effects of a binary treatment on education, measured by a test.
- Let's set up a simulation.
- 250 students. Ten classes of 25 students each. Observed over two years.
- First year has half good teachers and half bad.
- We want to estimate the effect of the intervention in year 2.
- Treatment is assigned randomly by **individual**

- Note: This setup usually demands an accounting of clustering, which I'm ignoring. Maybe I'll bring it back later in the semester when we discuss SUTVA.

Simulation

```
#Variables which govern the size of the simulation (and our causal effects)
nclass <- 5
nstudent <- 25
Eff <- 5
EffSD <- 3
# Simulate data
set.seed(1977)
Yr1ClassType <- rep(c(1,0),nclass*nstudent)
Yr2ClassType <- sample(Yr1ClassType,replace=FALSE)
Yr1Score <- rnorm(2*nclass*nstudent,76+Yr1ClassType*5,9)
# Fixed margins randomization
Trt  <- sample(Yr1ClassType,replace=FALSE)
# There is an independent effect of class type in each year
# Variance is different across class types in year 2
CtlOutcome <- rnorm(2*nclass*nstudent,Yr1Score+Yr2ClassType*3,9-Yr2ClassType*4)
# Treatment effect is random, but with expectation Eff
Yr2Obs <- CtlOutcome + Trt * rnorm(2*nclass*nstudent,Eff,EffSD)

summary(lm(Yr2Obs~Trt))$coefficients[2,]

##    Estimate  Std. Error     t value    Pr(>|t|)
## 4.307282238 1.558184168 2.764295984 0.006132827

summary(lm(Yr2Obs~Trt+Yr1Score))$coefficients[2,]

##     Estimate   Std. Error      t value     Pr(>|t|)
## 3.5206647114 1.0064194358 3.4982081884 0.0005553714

# We don't want the model-based SEs,
# we want the robust standard errors:
try(library('sandwich'),silent=TRUE)
try(library('lmtest'),silent=TRUE)
mod <- lm(Yr2Obs~Trt+Yr1Score)
coeftest(mod,vcovHC(mod,type='HC2'))["Trt",2]

## [1] 0.9997826
```

Plot Data

```
plot(jitter(Trt),Yr2Obs,axes=F,xlab="Treatment",ylab="Test Result (Yr 2)",col="grey")
axis(2)
axis(1,at=c(0,1))
# Calculate quantities for plotting CIs
mns <- tapply(Yr2Obs,Trt,mean)
# SEs could also be pulled from the linear models we fit above with:
ses <- tapply(Yr2Obs,Trt,function(x) sd(x)/sqrt(length(x)))
points(c(0,1),mns,col="red",pch=19)
# Note the loop so that I only write this code once
for(tr in unique(Trt)) {
  for(q in c(.25,.025)) {
    upr<-mns[as.character(tr)]+qnorm(1-q)*ses[as.character(tr)]
    lwr <- mns[as.character(tr)]-qnorm(1-q)*ses[as.character(tr)]
    segments(tr,upr,tr,lwr,lwd=(-4/log(q)))
  }
}
```

Partial Regression

- Can we make that plot a little more friendly?
- Let's residualize our outcome based on scores in the first period. This should remove a substantial amount of the variance in the outcome.
- A few things to check, though.

. . .

```
OutcomeRes <- residuals(lm(Yr2Obs~Yr1Score+0))
TrtRes <- residuals(lm(Trt~Yr1Score+0))
c(sd(OutcomeRes),sd(Yr2Obs))
```

```
## [1]  8.131035 12.481726
```

```
# Diagnostics
par(mfrow=c(1,4))
plot(Yr1Score,Yr2Obs)
plot(Yr1Score,jitter(Trt))
hist(OutcomeRes)
hist(TrtRes)
```

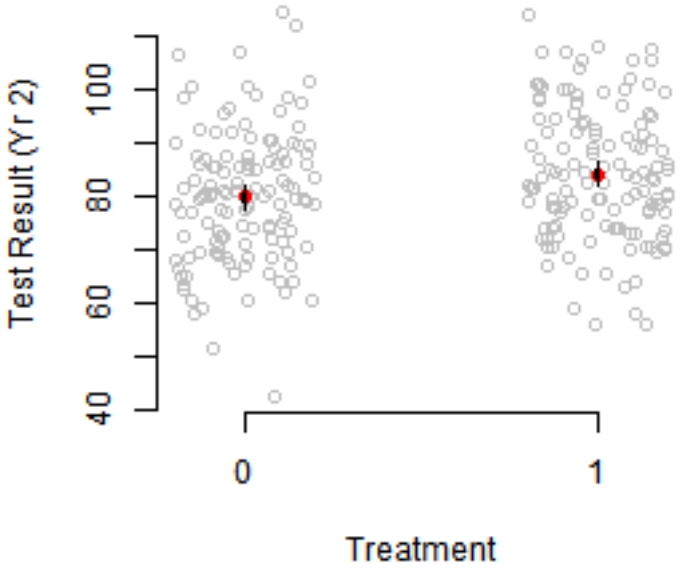

Figure 1:

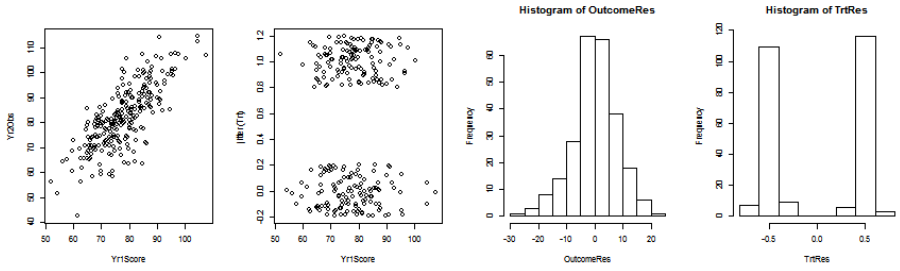

Figure 2:

Residualized Plot

```
par(mfrow=c(1,1))
plot(jitter(TrtRes),OutcomeRes,axes=F,xlab="Treatment (residuals)",ylab="Test Result (residu
axis(2)
axis(1)
# Pull information from the new bivariate model
mns<-coef(lm(OutcomeRes~TrtRes))
ses<-summary(lm(OutcomeRes~TrtRes))$coefficients[,2]
TrtResMns<-tapply(TrtRes,Trt,mean)
names(ses)<-names(mns)<-names(TrtResMns)
points(TrtResMns,mns,col="red",pch=19)
for(tr in names(TrtResMns)) {
  for(q in c(.25,.025)) {
    upr<-mns[tr]+qnorm(1-q)*ses[tr]
    lwr <- mns[tr]-qnorm(1-q)*ses[tr]
    segments(TrtResMns[tr],upr,TrtResMns[tr],lwr,lwd=(-4/log(q)))
  }
}
```

Effective Samples

- We're going to be investigating how to check the properties of your effective sample in regression.
- The key result is:
  $\hat{\rho}_{reg} \xrightarrow{p} \frac{E[w_i \rho_i]}{E[w_i]}$ where $w_i = (D_i - E[D_i|X_i])^2$
- We estimate these weights with:
  $\hat{w}_i = \hat{D}_i^2$ where D_i^2 is the ith squared residual.
- Because these estimates are "bad" for each unit, using them to reweight the sample is a bad idea.
- Instead, we just use them to get a sense for what the effective sample is by examining the weight allocated to particular strata.
- We will now explore how to do this.

Example paper

- We will be looking at Egan and Mullin (2012)
- This paper explores the effect of local weather variations on belief in global warming.
- Very cool paper! With an interesting randomized treatment.
- But what is the effective sample?

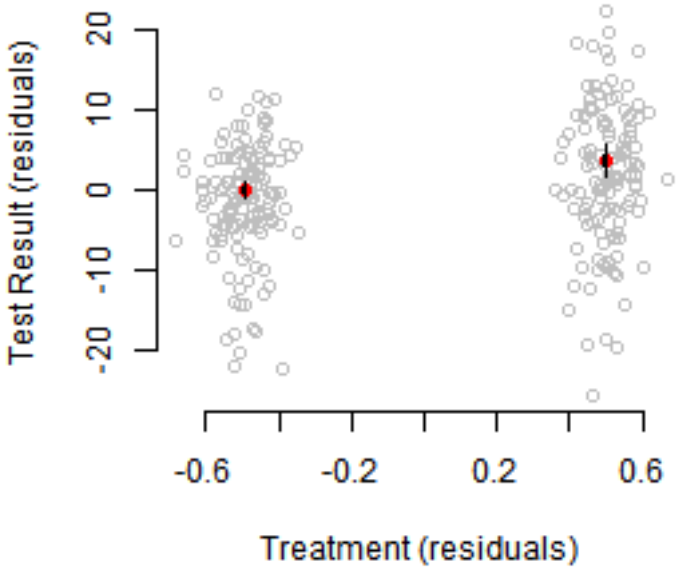
Test Result (residuals)
-20
-10
0
10
20
-0.6
-0.2
0.2
0.6
Treatment (residuals)

Figure 3:

- In other words, where is weather (conditional on covariates) most variable?
- That's what we'll explore.

Load in data

- Get the data from Pat's replication materials here.

. . .

```
require(foreign)
d <- read.dta("gwdataset.dta")
```

```
## Warning in read.dta("gwdataset.dta"): value labels ('q2') for 'jan07_q2'
## are missing
```

```
zips <- read.dta("zipcodetostate.dta")
zips<-unique(zips[,c("statenum","statefromzipfile")])
pops <- read.csv("population_ests_2013.csv")
pops$state <- tolower(pops$NAME)
d$getwarmord <- as.double(d$getwarmord)
# And estimate primary model of interest:
out<-lm(getwarmord~ddt_week+educ_hsless+educ_coll+educ_postgrad+educ_dk+party_rep+party_lear
```

Base Model

- We won't worry about standard errors yet.

. . .

```
summary(out)$coefficients[1:10,]
```

```
##                   Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)    1.945740062 0.771478843  2.5220913 1.169077e-02
## ddt_week       0.004857915 0.002475887  1.9620908 4.979656e-02
## educ_hsless    0.057740175 0.024483855  2.3582959 1.838991e-02
## educ_coll      0.021456581 0.027094559  0.7919147 4.284407e-01
## educ_postgrad  0.049387053 0.030734047  1.6069167 1.081236e-01
## educ_dk        0.102935825 0.250495224  0.4109293 6.811386e-01
## party_rep     -0.243900010 0.036626759 -6.6590661 2.992461e-11
## party_leanrep -0.092808607 0.041811888 -2.2196703 2.647719e-02
## party_leandem  0.147408845 0.039023948  3.7773944 1.599612e-04
## party_dem      0.175426658 0.035266683  4.9742886 6.725131e-07
```

Estimate D^2

- We can simply square the residuals of a partial regression to get D^2:

. . .

```
outD<-lm(ddt_week~educ_hsless+educ_coll+educ_postgrad+educ_dk+party_rep+party_leanrep+party_
D2 <- residuals(outD)^2
```

Effective Sample Statistics

- We can use these estimated weights for examining the sample.

. . .

```
compare_samples<- d[,c("wave","ddt_week","ddt_twoweeks","ddt_threeweeks","party_rep","attend
compare_samples <- apply(compare_samples,2,function(x) c(mean(x),sd(x),weighted.mean(x,D2),s
compare_samples <- t(compare_samples)
colnames(compare_samples) <- c("Nominal Mean","Nominal SD","Effective Mean", "Effective SD")
compare_samples
```

```
##                   Nominal Mean Nominal SD Effective Mean Effective SD
## wave                3.09693726  1.4252527     3.20788200    1.5609143
## ddt_week            3.83548593  5.9047249     5.11579140   10.8980228
## ddt_twoweeks        3.85505617  5.4572382     5.00137435    9.2262827
## ddt_threeweeks      3.96719696  4.7689594     5.10859485    8.4348180
## party_rep           0.29527208  0.4561989     0.28978321    0.4536617
## attend_1            0.11433244  0.3182383     0.12343459    0.3289354
## ideo_conservative   0.31132917  0.4630715     0.29325249    0.4552532
## age_1824            0.07195956  0.2584402     0.06881146    0.2531333
## educ_hsless         0.34151056  0.4742516     0.31219962    0.4633908
```

Effective Sample Maps

- But one of the most interesting things is to see this visually.
- Where in the US does the effective sample emphasize?
- To get at this, we'll use some tools in R that make this incredibly easy.
- In particular, we'll do this in ggplot2.

. . .

```r
# Effective sample by state
wt.by.state <- tapply(D2,d$statenum,sum)
wt.by.state <- wt.by.state/sum(wt.by.state)*100
wt.by.state <- cbind(D2=wt.by.state,statenum=names(wt.by.state))
data_for_map <- merge(wt.by.state,zips,by="statenum")
# Nominal Sample by state
wt.by.state <- tapply(rep(1,6726),d$statenum,sum)
wt.by.state <- wt.by.state/sum(wt.by.state)*100
wt.by.state <- cbind(Nom=wt.by.state,statenum=names(wt.by.state))
data_for_map <- merge(data_for_map,wt.by.state,by="statenum")
# Get correct state names
require(maps,quietly=TRUE)
data(state.fips)
data_for_map <- merge(state.fips,data_for_map,by.x="abb",by.y="statefromzipfile")
data_for_map$D2 <- as.double(as.character(data_for_map$D2))
data_for_map$Nom <- as.double(as.character(data_for_map$Nom))
data_for_map$state <- sapply(as.character(data_for_map$polyname),function(x)strsplit(x,":")
data_for_map$Diff <- data_for_map$D2 - data_for_map$Nom
data_for_map <- merge(data_for_map,pops,by="state")
data_for_map$PopPct <- data_for_map$POPESTIMATE2013/sum(data_for_map$POPESTIMATE2013)*100
data_for_map$PopDiffEff <- data_for_map$D2 - data_for_map$PopPct
data_for_map$PopDiffNom <- data_for_map$Nom - data_for_map$PopPct
data_for_map$PopDiff <- data_for_map$PopDiffEff - data_for_map$PopDiffNom
require(ggplot2,quietly=TRUE)
state_map <- map_data("state")
```

More setup

```r
plotEff <- ggplot(data_for_map,aes(map_id=state))
plotEff <- plotEff + geom_map(aes(fill=D2), map = state_map)
plotEff <- plotEff + expand_limits(x = state_map$long, y = state_map$lat)
plotEff <- plotEff + scale_fill_continuous("% Weight",limits=c(0,16),low="white", high="blac
plotEff <- plotEff + labs(title = "Effective Sample")
plotEff <- plotEff + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)

plotNom <- ggplot(data_for_map,aes(map_id=state))
plotNom <- plotNom + geom_map(aes(fill=Nom), map = state_map)
plotNom <- plotNom + expand_limits(x = state_map$long, y = state_map$lat)
```

```
plotNom <- plotNom + scale_fill_continuous("% Weight",limits=c(0,16),low="white", high="blac
plotNom <- plotNom + labs(title = "Nominal Sample")
plotNom <- plotNom + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)
```

And the maps

```
require(gridExtra,quietly=TRUE)
grid.arrange(plotNom,plotEff,ncol=2)
```

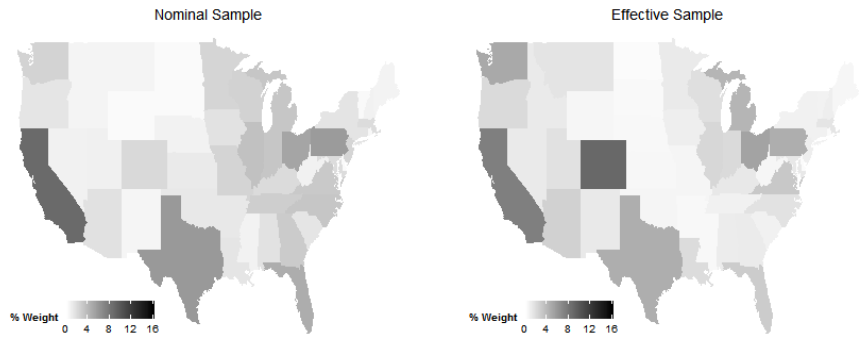

Figure 4:

Setup Comparison Plot

```
plotDiff <- ggplot(data_for_map,aes(map_id=state))
plotDiff <- plotDiff + geom_map(aes(fill=Diff), map = state_map)
plotDiff <- plotDiff + expand_limits(x = state_map$long, y = state_map$lat)
plotDiff <- plotDiff + scale_fill_gradient2("% Weight",low = "red", mid = "white", high = "b
plotDiff <- plotDiff + labs(title = "Effective Weight Minus Nominal Weight")
plotDiff <- plotDiff + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
```

```
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)
```

Difference in Weights

```
plotDiff
```

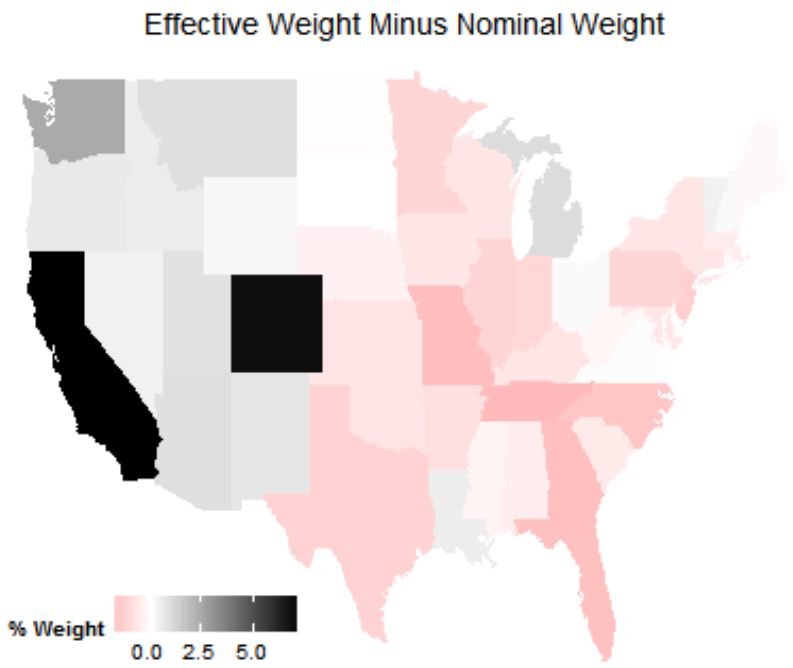

Figure 5:

Population Comparison

```
plotEff <- ggplot(data_for_map,aes(map_id=state))
plotEff <- plotEff + geom_map(aes(fill=PopDiffEff), map = state_map)
plotEff <- plotEff + expand_limits(x = state_map$long, y = state_map$lat)
plotEff <- plotEff + scale_fill_gradient2("% Weight",limits=c(-2,6),low="red",mid="white",hi
```

```
plotEff <- plotEff + labs(title = "Effective Sample")
plotEff <- plotEff + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)

plotNom <- ggplot(data_for_map,aes(map_id=state))
plotNom <- plotNom + geom_map(aes(fill=PopDiffNom), map = state_map)
plotNom <- plotNom + expand_limits(x = state_map$long, y = state_map$lat)
plotNom <- plotNom + scale_fill_gradient2("% Weight",limits=c(-2,6),low = "red",mid="white",
plotNom <- plotNom + labs(title = "Nominal Sample")
plotNom <- plotNom + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)
```

Population Comparison Plots

```
grid.arrange(plotNom,plotEff,ncol=2)
```

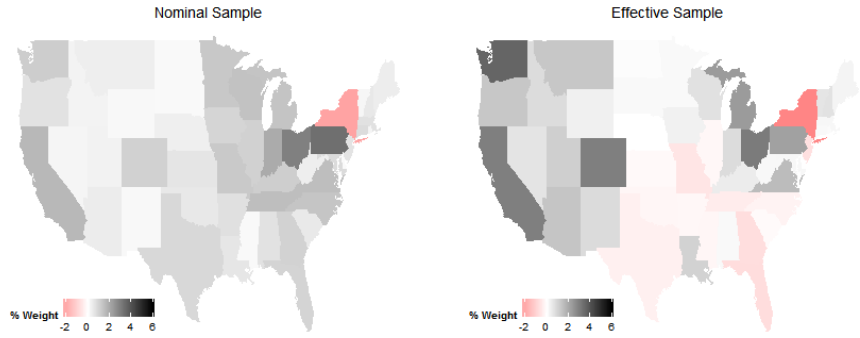

Figure 6:

Setup New Comparison Plot

```
plotDiff <- ggplot(data_for_map,aes(map_id=state))
plotDiff <- plotDiff + geom_map(aes(fill=PopDiff), map = state_map)
plotDiff <- plotDiff + expand_limits(x = state_map$long, y = state_map$lat)
plotDiff <- plotDiff + scale_fill_gradient2("% Weight",low = "red", mid = "white", high = "b
plotDiff <- plotDiff + labs(title = "Effective Weight Minus Nominal Weight")
plotDiff <- plotDiff + theme(
        legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(), plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank()
)
```

Plot Difference

```
plotDiff
```

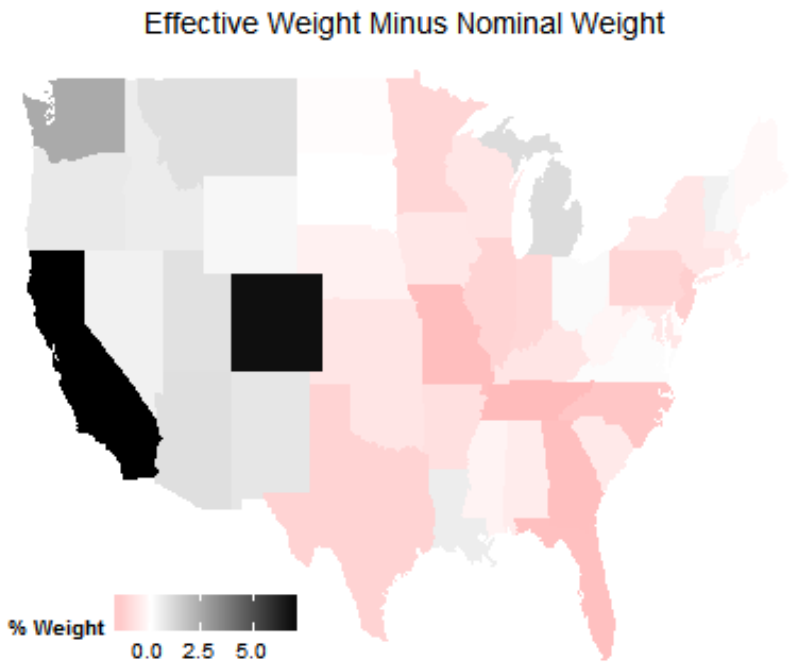

Figure 7: