

# Health Care Report

*Julie Ghekas*

*April 1, 2016*

## Setting up the data

For the first time reading in the data, the following commands should be given to R, replacing the appropriate locations for the mortality and population files:

```
#mort99=read.fwf('~Downloads/mort/MORT9913.txt',width=c(2,3,4,1,1,2,4,3,4))
#colnames(mort99)=c('state','county','year','racesex','hisp','age','ICD10cause','cause','deaths')
#pop99=read.fwf('~Downloads/pop9913.txt',width=c(2,3,4,1,1,rep(8,14),25,1))
#colnames(pop99)=c('state','county','year','racesex','hisp','birth','l1','a14','a59','a1014','a1519','a1914')
arf<-read.csv('~Downloads/arfarfsixten.csv',header=T)
colnames(arf)<-c('state','county','med610','medinc10','medinc06','pov10','pov06','ins10','ins06','unemp')
```

Reading in the datasets takes a while, so the first time, these should also be saved as a Rdata file, using the following commands:

```
#save(mort99,file='mort99.Rdata')
#save(pop99,file='pop99.Rdata')
```

Loading the mortality and population data in the future can then be done with the following commands much more quickly:

```
setwd('~')
load('mort99.Rdata')
load('pop99.Rdata')
```

The first time that R is using, the following packages need to be installed using the command `install.packages('package_name',dep=T)`. Following the first install, it should remain installed on the machine, and can be loaded using the library command.

```
library(reshape2)
library(car)
library(optmatch)
```

```
## Loading required package: digest
## You're loading optmatch, by B. Hansen and M. Fredrickson.
## The optmatch package makes essential use of D. P. Bertsekas
## and P. Tseng's RELAX-IV algorithm and code, as well as
## Bertsekas' AUCTION algorithm and code. Using the software
## to 'satisfy in any part commercial delivery requirements to
## government or industry' requires a special agreement with
## Dr. Bertsekas. For more information, enter
## relaxinfo() at the command line.
```

```
library(RItools)
```

## Massachusetts Replication

### Creating the Massachusetts Data

The form of the data needs to be modified to be in the form of analysis. This will be done below, calculating the appropriate percentages for the different variables needed and incorporating all of the variables into the same dataframe.

```
pop06=subset(pop99,pop99$year>=2001 & pop99$year<=2006)
pop06=subset(pop06,pop06$type==3)
pop06$a2034=pop06$a2024+pop06$a2534
popall=subset(pop99,pop99$type==3)
popall$a2034=popall$a2024+popall$a2534

agemat=aggregate(cbind(a2034,a3544,a4554,a5564)~county+state,sum,data=pop06)
ages=cbind(agemat[,1:2],100*t(apply(agemat[,3:6],1,prop.table)))

racesexmat=aggregate(cbind(count=a2034+a3544+a4554+a5564)~racesex+county+state,sum,data=pop06)
racesmat=racesexmat
racesmat$race=recode(racesexmat$racesex,"1:2=1; 3:4=2; 5:8=3")
racesmat$race[racesmat$race==1]="white"
racesmat$race[racesmat$race==2]="black"
racesmat$race[racesmat$race==3]="otherrace"
races=aggregate(count~race+county+state,sum,data=racesmat)
races=dcast(data=races,state+county~race)
```

```
## Using count as value column: use value.var to override.
```

```
race=cbind(races[,1:2],100*t(apply(races[,3:5],1,prop.table)))

sexmat=racesexmat
sexmat$sex=recode(racesexmat$racesex, "1=1; 2=2; 3=1; 4=2; 5=1; 6=2; 7=1; 8=2")
sexmat$sex[sexmat$sex==1]="male"
sexmat$sex[sexmat$sex==2]="female"
sexes=aggregate(count~sex+county+state,sum,data=sexmat)
sexes=dcast(data=sexes,state+county~sex)
```

```
## Using count as value column: use value.var to override.
```

```
sex=cbind(sexes[,1:2],100*t(apply(sexes[,3:4],1,prop.table)))

hispmat=aggregate(cbind(count=a2034+a3544+a4554+a5564)~hisp+county+state,sum,data=pop06)
hispmat$hisp=recode(hispmat$hisp, '2=1; 1=2; 9=2')
hispmat$hisp[hispmat$hisp==1]="hisp"
hispmat$hisp[hispmat$hisp==2]='nothisp'
hisps=aggregate(count~hisp+county+state,sum,data=hispmat)
hisps=dcast(data=hisps,state+county~hisp)
```

```
## Using count as value column: use value.var to override.
```

```
hisp=cbind(hisps[,1:2],100*t(apply(hisps[,3:4],1,prop.table)))

mort06=subset(mort99,mort99$year>=2001 & mort99$year<=2006)
mort06=subset(mort06,mort06$age>=09 & mort06$age<=13)
deaths=aggregate(deaths~county+state+year,sum,data=mort06)
pop06$pops=pop06$a2024+pop06$a2534+pop06$a3544+pop06$a4554+pop06$a5564
pops=aggregate(pops~county+state+year,sum,data=pop06)
popweights=aggregate(pops~county+state,sum,data=pop06)
total=merge(pops,deaths,by=c('year','state','county'))
total$mrates=total$deaths/total$pops*100000
totals=dcast(data=total,state+county~year,value.var='mrates')
colnames(totals)=c('state','county','mr01','mr02','mr03','mr04','mr05','mr06')

mortall=subset(mort99, mort99$age>=09 & mort99$age<=13)
deathall=aggregate(deaths~county+state+year, sum, data=mortall)
popall$pops=popall$a2024+popall$a2534+popall$a3544+popall$a4554+popall$a5564
popall=aggregate(pops~county+state+year,sum,data=popall)
totalall<-merge(popall,deathall,by=c('year','state','county'))
totalall$mrates=totalall$deaths/totalall$pops*100000
totalsall=dcast(data=totalall,state+county~year,value.var='mrates')
colnames(totalsall)=c('state','county','mr99','mr00','mr01','mr02','mr03','mr04','mr05','mr06','mr07','mr08','mr09')
totalsall[is.na(totalsall)]=0

model2data=merge(ages,race,by=c('state','county'))
model2data=merge(model2data,sex,by=c('state','county'))
model2data=merge(model2data,hisp,by=c('state','county'))
model2data=merge(model2data,totals,by=c('state','county'))
model2data$treat=0
model2data$treat[model2data$state==25]=1
model2data=merge(model2data,arf,by=c('state','county'))
model2data[is.na(model2data)]=0
model2data=merge(model2data,popweights,by=c('state','county'))
```

There are some assumptions made in the calculation of the model2data data frame. See the mass\_code for a full listing of those assumptions.

## Replicating the Analysis

```
model2=glm(treat~a2034+a3544+a4554+male+white+black+hisp+mr01+mr02+mr03+mr04+mr05+mr06+unemp06+pov06+in
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

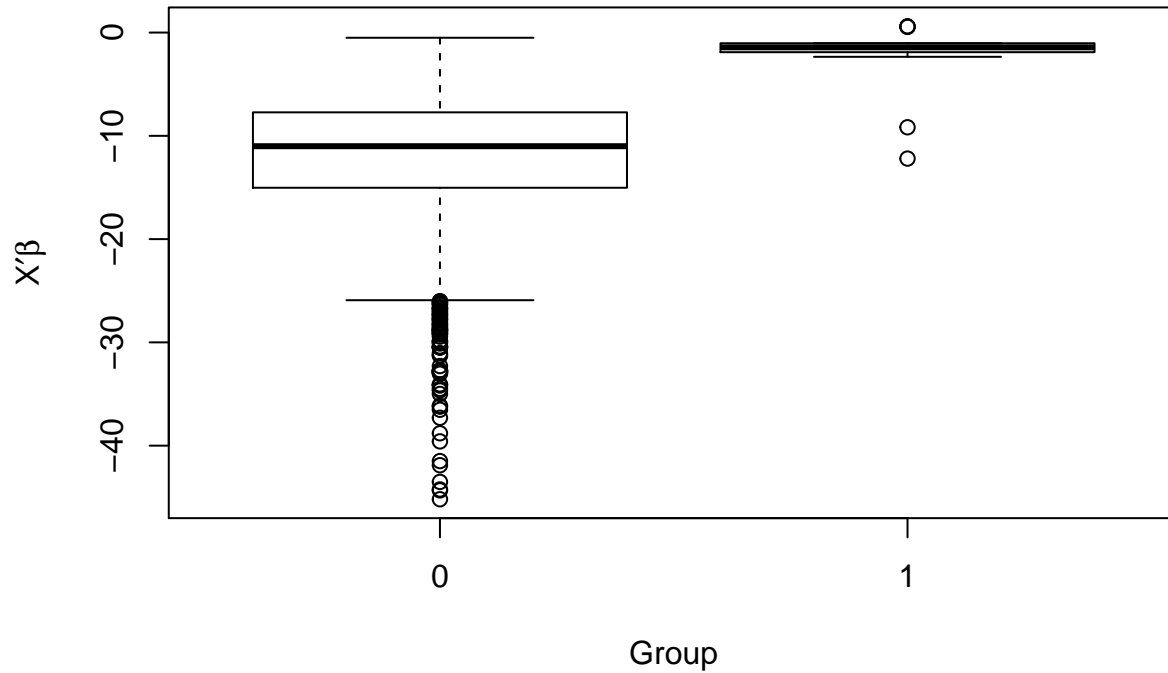
```
summary(model2)
```

```
##
## Call:
## glm(formula = treat ~ a2034 + a3544 + a4554 + male + white +
##       black + hisp + mr01 + mr02 + mr03 + mr04 + mr05 + mr06 +
```

```
##      unemp06 + pov06 + ins06 + medinc06, family = binomial(link = "logit"),
##      data = model2data)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -0.9724  -0.0295  -0.0057  -0.0007   4.9402
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  8.816e+01  2.781e+01   3.170 0.001527 **
## a2034        -1.847e-01  2.055e-01  -0.899 0.368855
## a3544        -5.422e-01  2.343e-01  -2.315 0.020628 *
## a4554        -3.791e-01  4.086e-01  -0.928 0.353464
## male         -1.179e+00  3.375e-01  -3.493 0.000478 ***
## white         5.924e-03  6.226e-02   0.095 0.924200
## black        -1.634e-02  8.356e-02  -0.196 0.844936
## hisp         8.318e-02  5.118e-02   1.625 0.104081
## mr01         -6.897e-03  8.493e-03  -0.812 0.416745
## mr02         6.682e-03  8.381e-03   0.797 0.425299
## mr03        -7.038e-05  8.776e-03  -0.008 0.993601
## mr04        -1.647e-03  7.388e-03  -0.223 0.823551
## mr05        -7.047e-03  8.022e-03  -0.878 0.379677
## mr06        -1.598e-03  7.965e-03  -0.201 0.841037
## unemp06       6.144e-01  2.726e-01   2.254 0.024218 *
## pov06        -6.528e-02  1.776e-01  -0.368 0.713138
## ins06        -6.502e-01  1.429e-01  -4.549 5.38e-06 ***
## medinc06      7.398e-05  4.785e-05   1.546 0.122091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 179.50  on 3139  degrees of freedom
## Residual deviance:  98.84  on 3122  degrees of freedom
## AIC: 134.84
##
## Number of Fisher Scoring iterations: 11
```

```
boxplot(model2)
```

## Overlap on fitted scores



```
fit.q3=summary(fitted(model2))[5]
cont.count=which(fitted(model2)>fit.q3)
mass.treat=subset(model2data,model2data$state==25)
mass.control=model2data[cont.count,]
mass.control=subset(mass.control,mass.control$state!=25)
mass.study=rbind(mass.treat,mass.control)
xBalance(treat~a2034+a3544+a4554+a5564+black+otherrace+white+male+hisp+mr01+mr02+mr03+mr04+mr05+mr06, d
```

##	strata	unstrat					
##	stat	treat=0	treat=1	adj.diff	adj.diff.null.sd	std.diff	z
## vars							
## a2034	30.67655	31.18590	0.50935	1.24579	0.11020	0.40886	
## a3544	25.32808	25.95372	0.62564	0.57195	0.29502	1.09386	
## a4554	25.55198	25.23238	-0.31960	0.53584	-0.16078	-0.59644	
## a5564	18.44339	17.62800	-0.81539	0.79840	-0.27542	-1.02128	
## black	5.80935	5.85978	0.05043	2.94990	0.00461	0.01710	
## otherrace	3.30422	3.83003	0.52581	2.06472	0.06864	0.25467	
## white	90.88644	90.31019	-0.57625	3.58359	-0.04334	-0.16080	
## male	49.65466	49.02370	-0.63096	0.34994	-0.48692	-1.80303	.
## hisp	3.97157	5.74790	1.77633	1.90472	0.25147	0.93259	
## mr01	329.61720	282.25257	-47.36463	32.36181	-0.39498	-1.46360	
## mr02	343.35009	292.24516	-51.10493	30.10817	-0.45828	-1.69738	.
## mr03	344.73335	293.84400	-50.88935	28.39131	-0.48405	-1.79243	.
## mr04	333.98580	285.36825	-48.61756	29.24064	-0.44888	-1.66267	.
## mr05	341.00690	287.00116	-54.00574	29.38173	-0.49643	-1.83807	.
## mr06	349.53323	294.24919	-55.28404	29.77528	-0.50148	-1.85671	.
## ---Overall Test---							
##	chisquare	df	p.value				
## unstrat	13.4	13	0.417				

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The xBalance shows a different balance when

## Kentucky Analysis

### Setting Up the Data

```
pop13=subset(pop99,pop99$year>=2005)
pop13=subset(pop13,pop13$type==3)
pop13$a2034=pop13$a2024+pop13$a2534

agemat=aggregate(cbind(a2034,a3544,a4554,a5564)~county+state,sum,data=pop13)
ages=cbind(agemat[,1:2],100*t(apply(agemat[,3:6],1,prop.table)))

racesexmat=aggregate(cbind(count=a2034+a3544+a4554+a5564)~racesex+county+state,sum,data=pop13)
racesmat=racesexmat
racesmat$race=recode(racesexmat$racesex,"1:2=1; 3:4=2; 5:8=3")
racesmat$race[racesmat$race==1]="white"
racesmat$race[racesmat$race==2]="black"
racesmat$race[racesmat$race==3]="otherrace"
races=aggregate(count~race+county+state,sum,data=racesmat)
races=dcast(data=races,state+county~race)
```

## Using count as value column: use value.var to override.

```
race=cbind(races[,1:2],100*t(apply(races[,3:5],1,prop.table)))

sexmat=racesexmat
sexmat$sex=recode(racesexmat$racesex, "1=1; 2=2; 3=1; 4=2; 5=1; 6=2; 7=1; 8=2")
sexmat$sex[sexmat$sex==1]="male"
sexmat$sex[sexmat$sex==2]='female'
sexes=aggregate(count~sex+county+state,sum,data=sexmat)
sexes=dcast(data=sexes,state+county~sex)
```

## Using count as value column: use value.var to override.

```
sex=cbind(sexes[,1:2],100*t(apply(sexes[,3:4],1,prop.table)))

hispmat=aggregate(cbind(count=a2034+a3544+a4554+a5564)~hisp+county+state,sum,data=pop13)
hispmat$hisp=recode(hispmat$hisp, "2=1; 1=2; 9=2")
hispmat$hisp[hispmat$hisp==1]="hisp"
hispmat$hisp[hispmat$hisp==2]='nothisp'
hisps=aggregate(count~hisp+county+state,sum,data=hispmat)
hisps=dcast(data=hisps,state+county~hisp)
```

## Using count as value column: use value.var to override.

```

hisp=cbind(hisps[,1:2],100*t(apply(hisps[,3:4],1,prop.table)))

mort13=subset(mort99,mort99$year>=2005)
mort13=subset(mort13,mort13$age>=09 & mort13$age<=13)
deaths=aggregate(deaths~county+state+year,sum,data=mort13)
pop13$pops=pop13$a2024+pop13$a2534+pop13$a3544+pop13$a4554+pop13$a5564
pops=aggregate(pops~county+state+year,sum,data=pop13)
popweights=aggregate(pops~county+state,sum,data=pop13)
total=merge(pops,deaths,by=c('year','state','county'))
total$mrates=total$deaths/total$pops*100000
totals=dcast(data=total,state+county~year,value.var='mrates')
colnames(totals)=c('state','county','mr05','mr06','mr07','mr08','mr09','mr10','mr11','mr12','mr13')
totals[is.na(totals)]=0

model3data=merge(ages,race,by=c('state','county'))
model3data=merge(model3data,sex,by=c('state','county'))
model3data=merge(model3data,hisp,by=c('state','county'))
model3data=merge(model3data,totals,by=c('state','county'))
model3data=merge(model3data,popweights,by=c('state','county'))
model3data=merge(model3data,arf,by=c('state','county'))

states.kycontrol=c('1','12','13','16','20','21','22','23','28','29','31','37','40','45','46','47','48',
length(states.kycontrol))

```

```
## [1] 21
```

```

model3dat=model3data
model3data=model3data[model3data$state %in% states.kycontrol,]
model3data$treat=0
model3data$treat[model3data$state==21]=1

```

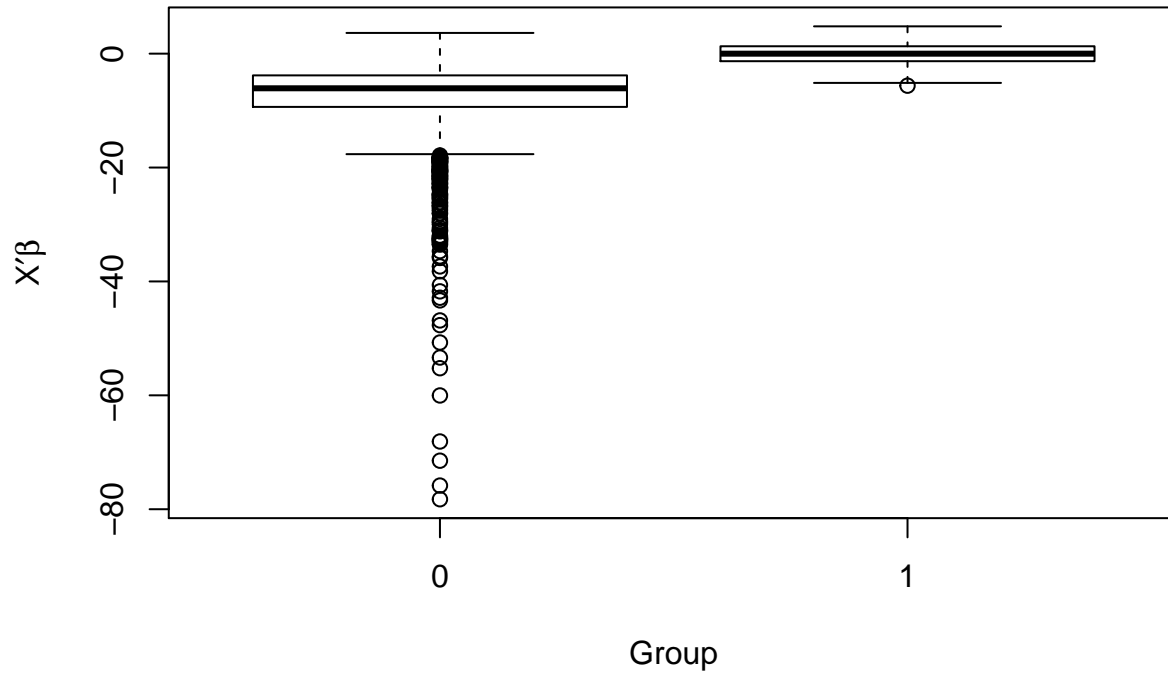
## Fitting the model and analysis

```
model3=glm(treat~a2034+a3544+a4554+male+white+black+hisp+mr05+mr06+mr07+mr08+mr09+mr10+pov10+medinc10+u
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
boxplot(model3)
```

## Overlap on fitted scores



```
ken.treat=subset(model3data,model3data$treat==1)
ken.control=model3data[which(fitted(model3)>summary(fitted(model3))[5]),]
ken.control=subset(ken.control,ken.control$treat!=1)
ken.study=rbind(ken.treat,ken.control)
xBalance(treat~a2034+a3544+a4554+a5564+black+otherrace+white+male+hisp+mr05+mr06+mr07+mr08+mr09+mr10+po
```

##	strata	unstrat						
##	stat	treat=0	treat=1	adj.diff	adj.diff.null.sd	std.diff	z	
## vars								
## a2034	3.05e+01	3.06e+01	1.23e-01	5.18e-01	2.52e-02	2.38e-01		
## a3544	2.25e+01	2.29e+01	3.14e-01	1.95e-01	1.71e-01	1.61e+00		
## a4554	2.52e+01	2.50e+01	-1.35e-01	2.15e-01	-6.63e-02	-6.26e-01		
## a5564	2.18e+01	2.15e+01	-3.03e-01	2.87e-01	-1.12e-01	-1.06e+00		
## black	6.24e+00	3.96e+00	-2.29e+00	8.07e-01	-3.03e-01	-2.84e+00	**	
## otherrace	1.14e+00	8.36e-01	-3.08e-01	6.94e-02	-4.80e-01	-4.44e+00	***	
## white	9.26e+01	9.52e+01	2.59e+00	8.16e-01	3.40e-01	3.18e+00	**	
## male	5.02e+01	5.02e+01	-2.22e-02	2.33e-01	-1.01e-02	-9.54e-02		
## hisp	2.47e+00	1.80e+00	-6.67e-01	2.01e-01	-3.56e-01	-3.32e+00	***	
## mr05	4.48e+02	4.87e+02	3.87e+01	1.51e+01	2.74e-01	2.57e+00	*	
## mr06	4.48e+02	4.84e+02	3.65e+01	1.42e+01	2.75e-01	2.58e+00	**	
## mr07	4.51e+02	4.88e+02	3.65e+01	1.42e+01	2.75e-01	2.58e+00	**	
## mr08	4.53e+02	4.96e+02	4.25e+01	1.39e+01	3.26e-01	3.05e+00	**	
## mr09	4.78e+02	5.43e+02	6.49e+01	1.61e+01	4.36e-01	4.04e+00	***	
## mr10	4.71e+02	5.41e+02	7.04e+01	1.61e+01	4.72e-01	4.37e+00	***	
## pov10	1.78e+01	2.11e+01	3.32e+00	6.34e-01	5.71e-01	5.24e+00	***	
## medinc10	4.06e+04	3.74e+04	-3.21e+03	9.80e+02	-3.50e-01	-3.27e+00	**	
## unemp10	1.00e+01	1.11e+01	1.08e+00	2.55e-01	4.57e-01	4.23e+00	***	
## ins10	2.26e+01	2.34e+01	8.04e-01	4.48e-01	1.91e-01	1.79e+00	.	
## ---Overall Test---								



```
##          chisquare df  p.value
## unstrat          119 17 2.19e-17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that the background balance is pretty poor when using the top quartile of the possible control counties. Instead, I went through a process of finding a match that has good background comparability. I went through trying to find a subset of the controls based solely on the propensity scores, but I found that the control group did not have a good balance when it was based on a cutoff within the propensity score. I tried matching based on specific variables as well. The match that I like the best is full matching while including 1/4 of the potential control counties, in keeping with the Sommers analysis (using 1/4 of the control counties). This provided a good balance and a good control group. I thought about further restricting the full matching, which can be used to increase sample size. I think that these modifications only complicate the model more and do not help with the effective sample size enough to justify the complications. For more details, see the `ky_code` file.

```
fm3<-fullmatch(model3,data=model3data,omit.fraction=3/4)
summary(fm3)
```

```
## Structure of matched sets:
## 5+:1  4:1  2:1  1:1  1:2  1:3  1:4  1:5+  0:1
##      4      1      6      17      5      7      5      28 1281
## Effective Sample Size: 110
## (equivalent number of matched pairs).
##
## sum(matched.distances)=7.32
## (within 1.58 of optimum).
## Percentiles of matched distances:
##      0%      50%      95%     100%
## 1.83e-05 6.55e-03 4.39e-02 3.05e-01
```

```
xBalance(treat~a2034+a3544+a4554+a5564+black+otherrace+white+male+hispanic+mr05+mr06+mr07+mr08+mr09+mr10+pov10)
```

##	strata	Unadj				fm3			
##	stat	adj.diff	std.diff	z		adj.diff	std.diff	z	
## vars									
## a2034		-4.20e-01	-7.02e-02	-7.43e-01		2.93e-01	4.90e-02	4.30e-01	
## a3544		1.17e+00	5.73e-01	6.01e+00 ***		-5.89e-02	-2.89e-02	-2.24e-01	
## a4554		-2.01e-01	-7.70e-02	-8.15e-01		-1.10e-01	-4.21e-02	-3.79e-01	
## a5564		-5.47e-01	-1.45e-01	-1.53e+00		-1.24e-01	-3.27e-02	-3.39e-01	
## black		-9.54e+00	-5.69e-01	-5.97e+00 ***		2.54e-01	1.52e-02	2.44e-01	
## otherrace		-2.31e+00	-3.46e-01	-3.65e+00 ***		1.28e-02	1.92e-03	1.56e-01	
## white		1.19e+01	6.85e-01	7.15e+00 ***		-2.67e-01	-1.54e-02	-2.58e-01	
## male		-4.42e-01	-1.35e-01	-1.43e+00		2.14e-01	6.56e-02	7.53e-01	
## hisp		-6.95e+00	-5.13e-01	-5.39e+00 ***		4.33e-02	3.19e-03	1.89e-01	
## mr05		5.21e+01	3.41e-01	3.60e+00 ***		-4.32e+00	-2.83e-02	-2.43e-01	
## mr06		5.29e+01	3.57e-01	3.76e+00 ***		-4.94e+00	-3.33e-02	-2.90e-01	
## mr07		5.72e+01	3.96e-01	4.17e+00 ***		-4.21e+00	-2.92e-02	-2.45e-01	
## mr08		5.96e+01	4.04e-01	4.26e+00 ***		-6.79e+00	-4.60e-02	-4.07e-01	
## mr09		1.00e+02	6.47e-01	6.76e+00 ***		1.01e+00	6.51e-03	5.52e-02	
## mr10		1.01e+02	6.60e-01	6.89e+00 ***		-3.42e+00	-2.24e-02	-1.92e-01	
## pov10		3.18e+00	4.89e-01	5.14e+00 ***		1.10e-01	1.69e-02	1.48e-01	

```
## medinc10      -3.96e+03 -4.01e-01 -4.23e+00 ***      3.26e+01  3.30e-03  2.66e-02
## unemp10       2.19e+00  7.10e-01  7.40e+00 ***      7.31e-02  2.37e-02  2.33e-01
## ins10        -1.08e+00 -1.76e-01 -1.86e+00 .      -3.83e-03 -6.21e-04 -7.00e-03
## ---Overall Test---
##      chisquare df  p.value
## Unadj    400.62 17 1.37e-74
## fm3       1.18 17 1.00e+00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
stratumStructure(fm3)
```

```
## 19:1  9:1  8:1  6:1  4:1  2:1  1:1  1:2  1:3  1:4  1:5  1:6  1:7  1:8  1:9
##      1      1      1      1      1      6      17      5      7      5      4      3      2      4      1
## 1:10 1:11 1:13 1:14 1:15 1:17 1:18 1:20 1:21 1:22 1:26 1:29 0:1
##      1      1      1      1      1      2      1      1      1      2      1      1 1281
## attr(,"comparable.num.matched.pairs")
## [1] 109.6014
```

## Tennessee Analysis

### Setting Up the Data

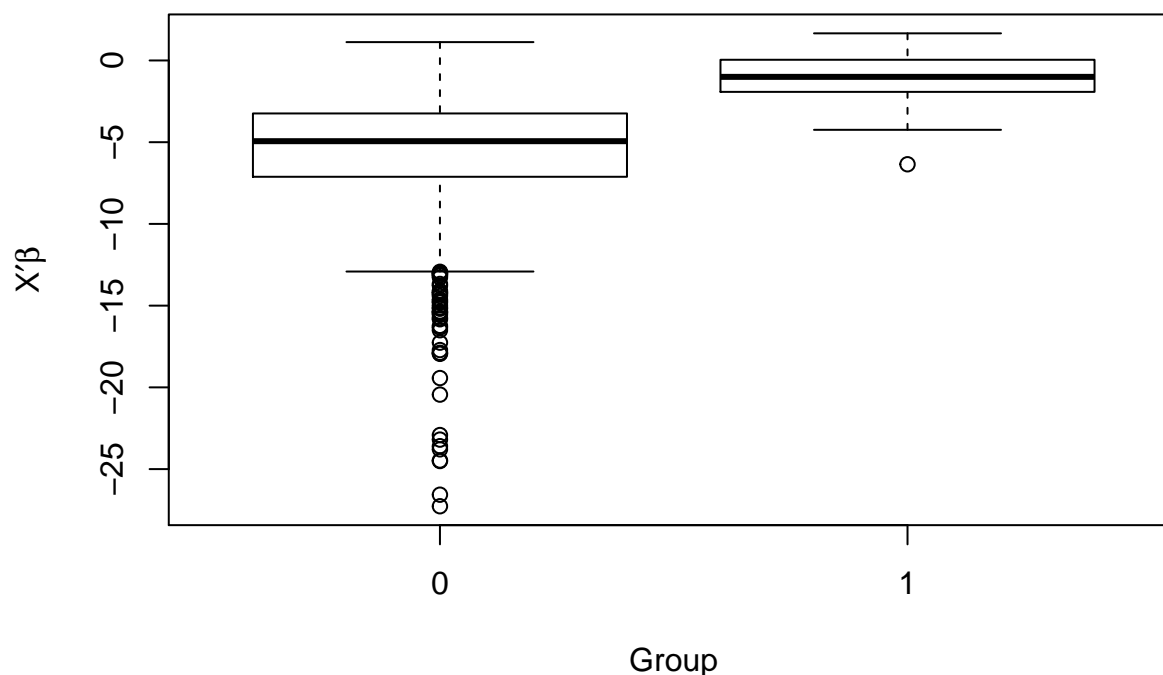
The data setup from Kentucky can be used, with a modification of what counties we use

```
model4data=model3dat
model4data$treat=0
model4data$treat[model4data$state==47]=1
states.tncontrol=c('4','5','6','8','9','10','17','18','19','21','24','25','26','27','30','32','33','34')
model4data=model4data[model4data$state %in% states.tncontrol,]
```

### Analysis of the Data

```
model4=glm(treat~a2034+a3544+a4554+male+white+black+hispanic+mr05+mr06+mr07+mr08+mr09+mr10+pov10+medinc10+unemp10+ins10)
boxplot(model4)
```

## Overlap on fitted scores



Again, I went through the same process of trying to replicate the Sommers analysis directly, using the top quartile of propensity scores as a control group. The balance did not seem, especially balanced. Since I selected full matching for the Kentucky analysis, I chose to continue down that route for the Tennessee analysis, proceeding as closely as possible to the Kentucky analysis.

```
fm22<-fullmatch(model4,data=model4data,omit.fraction=3/4)
stratumStructure(fm22)
```

```
## 8:1 4:1 3:1 2:1 1:1 1:2 1:3 1:4 1:5 1:6 1:7 1:8 1:9 1:10 1:11
## 1 1 2 1 21 8 15 9 4 2 1 1 3 2 3
## 1:12 1:17 1:21 1:26 0:1
## 3 1 1 1 1050
## attr("comparable.num.matched.pairs")
## [1] 115.6995
```

```
xBalance(treat~a2034+a3544+a4554+a5564+black+otherrace+white+male+hisp+mr05+mr06+mr07+mr08+mr09+mr10+po
```

	strata	Unadj			fm22		
	stat	adj.diff	std.diff	z	adj.diff	std.diff	z
## vars							
## a2034		1.18e-01	2.24e-02	2.11e-01	8.63e-02	1.64e-02	1.95e-01
## a3544		1.19e+00	6.02e-01	5.62e+00 ***	-2.16e-02	-1.09e-02	-1.17e-01
## a4554		-1.04e+00	-4.50e-01	-4.22e+00 ***	-4.69e-02	-2.04e-02	-2.92e-01
## a5564		-2.73e-01	-7.14e-02	-6.74e-01	-1.78e-02	-4.65e-03	-4.68e-02
## black		3.22e+00	4.09e-01	3.84e+00 ***	1.03e-01	1.30e-02	6.79e-02
## otherrace		-2.37e+00	-3.55e-01	-3.34e+00 ***	1.36e-02	2.04e-03	9.40e-02
## white		-8.55e-01	-8.20e-02	-7.73e-01	-1.16e-01	-1.11e-02	-7.39e-02
## male		-7.15e-01	-2.75e-01	-2.58e+00 **	2.11e-01	8.12e-02	6.56e-01

```
## hisp          -4.02e+00 -3.75e-01 -3.52e+00 ***      4.49e-02  4.18e-03  7.72e-02
## mr05          1.33e+02  1.05e+00  9.58e+00 ***      7.03e+00  5.53e-02  4.49e-01
## mr06          1.31e+02  1.03e+00  9.46e+00 ***      2.58e+00  2.04e-02  1.63e-01
## mr07          1.27e+02  9.53e-01  8.76e+00 ***      1.36e+00  1.02e-02  8.41e-02
## mr08          1.22e+02  9.13e-01  8.40e+00 ***      4.37e+00  3.26e-02  2.71e-01
## mr09          1.35e+02  9.83e-01  9.02e+00 ***     -1.90e-01 -1.38e-03 -1.09e-02
## mr10          1.36e+02  9.97e-01  9.14e+00 ***      5.62e+00  4.13e-02  3.35e-01
## pov10         4.37e+00  7.88e-01  7.30e+00 ***      5.58e-01  1.01e-01  7.85e-01
## medinc10      -7.34e+03 -6.69e-01 -6.23e+00 ***     -6.82e+02 -6.21e-02 -6.64e-01
## unemp10       1.84e+00  6.06e-01  5.66e+00 ***      3.22e-02  1.06e-02  1.08e-01
## ins10         1.75e+00  3.09e-01  2.91e+00 **       1.25e-01  2.20e-02  2.47e-01
## ---Overall Test---
##           chisquare df  p.value
## Unadj      212.85 17 7.36e-36
## fm22        4.39 17 9.99e-01
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fm22)
```

```
## Structure of matched sets:
## 5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5+  0:1
##    1    1    2    1   21    8   15    9   22 1050
## Effective Sample Size: 116
## (equivalent number of matched pairs).
##
## sum(matched.distances)=3.63
## (within 1.31 of optimum).
## Percentiles of matched distances:
##      0%      50%      95%     100%
## 1.59e-05 6.39e-03 1.83e-02 1.99e-01
```