replication

Helen Simpson

2/18/2019

```
library(ri)
library(RItools)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(xtable)
library(effects)
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(RColorBrewer)

options(scipen = 999)  ##set for non-scientific notaion output

##Load data
dat.all = read.csv('dataverse_files/pnas_data.csv')
dat.t1 = read.csv('dataverse_files/t1_data.csv')
dat.all.prime = read.csv('dataverse_files/prime_data.csv')
###data loading for faces graphic
conf.dat = read.csv('dataverse_files/confederate_face_data.csv')
hisp.dat = read.csv('dataverse_files/hispanic_face_data.csv')
white.dat = read.csv('dataverse_files/white_face_data.csv')
```

```
##balance_check.r
###check to see if randomization created balanced samples in Enos experiment
####RdE September 2012
#first output is table 2

#test for balance of demographic characteristics between treatment and control groups
out.balance.test = xBalance(fmla = treatment ~ liberal+republican+obama.disapprove+ride.everyday+voted.
print(out.balance.test)
```

```
##                  strata        strat
##                  stat    treatment=0  treatment=1     std.diff         z
## vars
## liberal                      0.46597      0.46904      0.00614      0.02995
## republican                   0.16723      0.18660      0.05014      0.24252
## obama.disapprove             0.27113      0.29404      0.04956      0.23745
## ride.everyday                0.84976      0.89973      0.15158      0.72286
```

```
## voted.2010              0.76523      0.66018      -0.23795     -1.12124
## romney.voter            0.24430      0.21542      -0.06671     -0.33702
## Hispanics.x             0.05816      0.04865      -0.07075     -0.33457
## age                     44.66037     40.42895     -0.35326     -1.63408
## residency.new           8.21673      7.07118      -0.19139     -0.90718
## hispanic.new            0.02820      0.05103      0.11491      0.58292
## college                 0.88599      0.86472      -0.06461     -0.30233
## income.new              144724.04374 138094.39975 -0.08747     -0.44455
## male                    0.60446      0.60166      -0.00560     -0.02692
## white                   0.90963      0.83356      -0.23850     -1.24865
## ---Overall Test---
##       chisquare df p.value
## strat       6.6 14   0.949
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```
xtable.out.balance.text = xtable(out.balance.test)
print(xtable.out.balance.text, file = 'balance_table.tex',
    floating = FALSE)

#test for balance of demographic characteristics between population and those who didn't take second su
missing.balance.test = xBalance(fmla = missing ~ liberal+republican+obama.disapprove+ride.everyday+voted
print(missing.balance.test)
```

```
##                  strata      unstrat
##                  stat      missing=0    missing=1     std.diff         z
## vars
## liberal                  0.43902      0.39437      -0.09055     -0.73441
## republican               0.17886      0.16901      -0.02610     -0.21068
## obama.disapprove         0.30081      0.28873      -0.02636     -0.21479
## ride.everyday            0.87805      0.78873      -0.24022     -1.92758     .
## voted.2010               0.73984      0.57042      -0.35884     -2.87637     **
## romney.voter             0.24390      0.27465      0.06993      0.56771
## Hispanics                1.21951      1.20423      -0.02483     -0.21236
## age                      25.54472     21.07746     -0.37290     -2.98750     **
## residency.new            7.30634      7.48014      0.03084      0.24981
## hispanic.new             0.04065      0.04225      0.00746      0.06513
## gender                   1.43902      1.38028      -0.11915     -0.96870
## college                  0.87805      0.84507      -0.09182     -0.77101
## us.born                  0.82114      0.76761      -0.13264     -1.06961
## income.new               145719.51220 136059.85915 -0.12932     -1.04703
## male                     0.56098      0.61972      0.11915      0.96870
## ---Overall Test---
##          chisquare df p.value
## unstrat      20.4 14   0.119
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```
#test for balance between outcome questions given demographic effects between complete/ missing groups
missing.balance.test = xBalance(fmla = missing ~ numberim+Remain+Englishlan+liberal+republican+obama.dis
print(missing.balance.test)
```

```
##                  strata      unstrat
##                  stat      missing=0    missing=1     std.diff         z
## vars
```

```
## numberim                2.96721      3.20423      0.22851      1.85208      .
## Remain                  2.77049      2.90845      0.09909      0.80204
## Englishlan              2.48361      2.29577      -0.13347     -1.07965
## liberal                 0.43443      0.39437      -0.08122     -0.65786
## republican              0.18033      0.16901      -0.02999     -0.24116
## obama.disapprove        0.30328      0.28873      -0.03174     -0.25778
## ride.everyday           0.88525      0.78873      -0.25958     -2.09388     *
## voted.2010              0.73770      0.57042      -0.35432     -2.83147     **
## romney.voter            0.24590      0.27465      0.06539      0.52898
## Hispanics               1.22131      1.20423      -0.02775     -0.23644
## age                     25.46721     21.07746     -0.36643     -2.92900     **
## residency.new           7.30038      7.48014      0.03190      0.25733
## hispanic.new            0.04098      0.04225      0.00591      0.05138
## gender                  1.43443      1.38028      -0.10982     -0.89169
## college                 0.87705      0.84507      -0.08904     -0.74482
## us.born                 0.82787      0.76761      -0.14931     -1.20795
## income.new              144864.75410 136059.85915 -0.11788     -0.95437
## male                    0.56557      0.61972      0.10982      0.89169
## ---Overall Test---
##         chisquare df p.value
## unstrat      23.9 17   0.123
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```r
#test for balance in treatment effect between complete/ missing groups??
missing.balance.test = xBalance(fmla = missing ~ treatment, data = dat.t1, report = c("std.diffs","z.sc
print(missing.balance.test)
```

```
##           strata   unstrat
##           stat   missing=0 missing=1 std.diff    z
## vars
## treatment            0.447     0.524    0.153   1.253
## ---Overall Test---
##         chisquare df p.value
## unstrat      1.57  1    0.21
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```r
##main_results.r
####primary randomization inferece
#this makes table 1
#not clear on why there are two p values

###inference
repeats = c("numberim","Remain","Englishlan")

x.names = paste(repeats,".x",sep="")
y.names = paste(repeats,".y",sep="")

covariates = c('line.x')

#splitting data into wait in car/ wait on platform
###treated first
final.mat = matrix(nrow = 0, ncol = 8)
```

```r
subsets = c('all','no.car')

cat('beginning inference \n')
```

```
## beginning inference
```

```r
for(subset in subsets){

    out.mat = matrix(nrow = length(repeats), ncol = 8)

    if(subset == 'all'){
        dat.subset = dat.all
        }
    if(subset ==  'no.car'){
        dat.subset = dat.all[dat.all$habits != 1,]
        }


    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        dat.use = dat.subset[is.na(dat.subset$Y) == F,]

        #calcuate values for table 1
        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)
        y.mean = mean(dat.use$y.new,na.rm = T)
        y.treat = mean(dat.use$y.new[dat.use$treatment==1],na.rm = T)

        station.treatment.table = table(dat.use$station,dat.use[,z.variable])
        no.control.stations = names(which(station.treatment.table[,1] == 0))
        no.treatment.stations = names(which(station.treatment.table[,2] == 0))
        dat.use = dat.use[!dat.use$station%in%c(no.control.stations,no.treatment.stations),]


        dat.use$station = factor(dat.use$station)
        dat.use$treated_unit = factor(dat.use$treated_unit)
        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treated_u
        probs = genprobexact(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treate

        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)
        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = 0)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

    #table formatting
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
```

```
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = disp$greater.p.value
        out.mat[j,6] = disp$lesser.p.value
        out.mat[j,7] = x.sd
        out.mat[j,8] = x.mean
        }
    final.mat = rbind(final.mat,out.mat)
    }

final.mat = as.data.frame(final.mat)
colnames(final.mat) = c('variable','subset','N','ate','greater.p.value','lesser.p.value','x.sd','x.mean
print(final.mat)
```

```
##     variable subset   N                ate greater.p.value lesser.p.value
## 1   numberim    all 106 0.0898125707655309       0.0078125      0.9921875
## 2     Remain    all 109 0.0727315441205261        0.015625       0.984375
## 3 Englishlan    all 109 0.0299129031275695      0.26953125     0.73046875
## 4   numberim no.car  98 0.0825966646148233      0.01171875     0.98828125
## 5     Remain no.car 100 0.0983554695104804        0.015625       0.984375
## 6 Englishlan no.car 100 0.0433034458889442      0.15234375     0.84765625
##                x.sd            x.mean
## 1 0.271834460776405 0.489130434782609
## 2 0.362030665966436 0.440677966101695
## 3 0.363553015812534  0.61864406779661
## 4 0.276280405627494  0.47877358490566
## 5   0.3524195653157 0.412037037037037
## 6 0.369610515536121 0.599537037037037
```

```
final.mat.main = final.mat ##final.mat for output creation later
```

```
##subset_inference.r
#not quite sure what this does or how it corresponds to the results in the paper.
#it seems to be using randomization?
#it's not clear where all the variables are coming from


repeats = c("numberim","Remain","Englishlan")

x.names = paste(repeats,".x",sep="")
y.names = paste(repeats,".y",sep="")

covariates = c('line.x')

var.names = c('Number of immigrants be increased?','Children of undocumented be allowed to stay?','Engli

#not sure what this does. What are ta and tb?
##dose response estimates comparison
final.mat = matrix(nrow = 0, ncol = 8)
subsets = c('ta','tb')
cat('beginning dose response inference \n')
```

```
## beginning dose response inference
```

```
for(subset in subsets){

    out.mat = matrix(nrow = length(repeats), ncol = 8)
    if(subset == 'ta'){
        dat.subset = dat.all[dat.all$t.time %in% c('t2a','t4a'),]
        }
    if(subset == 'tb'){
        dat.subset = dat.all[dat.all$t.time %in% c('t2b','t4b'),]
        }
    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        dat.use = dat.subset[is.na(dat.subset$Y) == F,]

        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)

        station.treatment.table = table(dat.use$station,dat.use[,z.variable])
        no.control.stations = names(which(station.treatment.table[,1] == 0))
        no.treatment.stations = names(which(station.treatment.table[,2] == 0))
        dat.use = dat.use[!dat.use$station%in%c(no.control.stations,no.treatment.stations),]

        dat.use$station = factor(dat.use$station)
        dat.use$treated_unit = factor(dat.use$treated_unit)
        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treated_u
        probs = genprobexact(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treate
        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)
        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = ate)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

        #format table
        ##fill matrix
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = x.mean
        out.mat[j,6] = x.sd
        out.mat[j,7] = disp$quantile[1]
        out.mat[j,8] = disp$quantile[2]
        }
        final.mat = rbind(final.mat,out.mat)
    }
final.mat = as.data.frame(final.mat)
colnames(final.mat) = c('variable','subset','N','ate','x.mean','x.sd','quantile.lower','quantile.upper')
print(final.mat)
```

```
##     variable subset  N                ate            x.mean
## 1   numberim     ta 51  0.128398004021747 0.504385964912281
## 2     Remain     ta 51  0.113297832725106 0.451754385964912
## 3 Englishlan     ta 51 0.0506628435242422  0.62719298245614
## 4   numberim     tb 51 0.0702224052718288 0.474137931034483
## 5     Remain     tb 54 0.0352650494159929 0.430327868852459
## 6 Englishlan     tb 54   0.01740790655885 0.610655737704918
##                x.sd      quantile.lower     quantile.upper
## 1 0.265128119594091  0.0359717401834773  0.220824267860017
## 2 0.385097959740754 -0.0271679675403221  0.253763632990535
## 3 0.341094636982826 -0.0561867063120087  0.157512393360493
## 4 0.279762019686022 -0.0487068209385605  0.189151631482218
## 5 0.341964750684018 -0.0553158692722372  0.125845968104223
## 6  0.38601756804779 -0.0639857987111565 0.0988016118288565
```

```
final.mat.dose = final.mat ##mat for creating graph later

#this does the same thing as above for different ideological subgroups
#not sure where in paper this appears (if at all)
##ideology comparison
final.mat = matrix(nrow = 0, ncol = 8)
subsets = c('liberals.only','moderates.only','conservatives.only')
cat('beginning ideology subset inference \n')
```

```
## beginning ideology subset inference
```

```
for(subset in subsets){

    out.mat = matrix(nrow = length(repeats), ncol = 8)
    if(subset == 'liberals.only'){
        dat.subset = dat.all[dat.all$ideology.x %in% c(1,2),]
        }
    if(subset == 'conservatives.only'){
        dat.subset = dat.all[dat.all$ideology.x %in% c(4,5),]
        }
    if(subset == 'moderates.only'){
        dat.subset = dat.all[dat.all$ideology.x == 3,]
        }
    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)

        station.treatment.table = table(dat.use$station,dat.use[,z.variable])
        no.control.stations = names(which(station.treatment.table[,1] == 0))
        no.treatment.stations = names(which(station.treatment.table[,2] == 0))
        dat.use = dat.use[!dat.use$station%in%c(no.control.stations,no.treatment.stations),]

        dat.use$station = factor(dat.use$station)
        dat.use$treated_unit = factor(dat.use$treated_unit)
```

```
        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treated_
        probs = genprobexact(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treat
        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)

        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = ate)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

        ##fill matrix
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = x.mean
        out.mat[j,6] = x.sd
        out.mat[j,7] = disp$quantile[1]
        out.mat[j,8] = disp$quantile[2]

        }
    final.mat = rbind(final.mat,out.mat)
    }
final.mat = as.data.frame(final.mat)
colnames(final.mat) = c('variable','subset','N','ate','x.mean','x.sd','quantile.lower','quantile.upper'
print(final.mat)
```

```
##     variable             subset  N              ate            x.mean
## 1   numberim      liberals.only 54 0.01740790655885 0.587962962962963
## 2     Remain      liberals.only 54 0.01740790655885 0.587962962962963
## 3 Englishlan      liberals.only 54 0.01740790655885 0.587962962962963
## 4   numberim     moderates.only 54 0.01740790655885 0.587962962962963
## 5     Remain     moderates.only 54 0.01740790655885 0.587962962962963
## 6 Englishlan     moderates.only 54 0.01740790655885 0.587962962962963
## 7   numberim conservatives.only 54 0.01740790655885 0.587962962962963
## 8     Remain conservatives.only 54 0.01740790655885 0.587962962962963
## 9 Englishlan conservatives.only 54 0.01740790655885 0.587962962962963
##                x.sd      quantile.lower     quantile.upper
## 1 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 2 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 3 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 4 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 5 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 6 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 7 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 8 0.392011646099765 -0.0639857987111565 0.0988016118288565
## 9 0.392011646099765 -0.0639857987111565 0.0988016118288565
```

```
final.mat.ideology = final.mat ##for graph later

#not sure what this is or what it corresponds to in the paper (if at all)
##friends comparison
final.mat = matrix(nrow = 0, ncol = 8)
subsets = c('low.friends','high.friends','middle.friends')
```

```r
cat('beginning friends response inference \n')
```

```
## beginning friends response inference
```

```r
for(subset in subsets){

    out.mat = matrix(nrow = length(repeats), ncol = 8)
    if(subset == 'low.friends'){
        dat.subset = dat.all[dat.all$Friends.x == 0,]
        }
    if(subset == 'high.friends'){
        dat.subset = dat.all[dat.all$Friends.x >= 5,]
        }
    if(subset == 'middle.friends'){
        dat.subset = dat.all[dat.all$Friends.x > 0 & dat.all$Friends.x < 5,]
        }

    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        dat.use = dat.subset[is.na(dat.subset$Y) == F,]


        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)

        station.treatment.table = table(dat.use$station,dat.use[,z.variable])
        no.control.stations = names(which(station.treatment.table[,1] == 0))
        no.treatment.stations = names(which(station.treatment.table[,2] == 0))
        dat.use = dat.use[!dat.use$station%in%c(no.control.stations,no.treatment.stations),]

        dat.use$station = factor(dat.use$station)
        dat.use$treated_unit = factor(dat.use$treated_unit)
        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treated_u
        probs = genprobexact(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treate

        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)
        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = ate)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

        ##fill matrix
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = x.mean
        out.mat[j,6] = x.sd
```

```
        out.mat[j,7] = disp$quantile[1]
        out.mat[j,8] = disp$quantile[2]

        #print(disp)
        }
    final.mat = rbind(final.mat,out.mat)
    }
    final.mat = as.data.frame(final.mat)
    colnames(final.mat) = c('variable','subset','N','ate','x.mean','x.sd','quantile.lower','quantile.up
    print(final.mat)
```

```
##     variable         subset  N                 ate            x.mean
## 1   numberim    low.friends 23   0.142694063926941 0.492857142857143
## 2     Remain    low.friends 24   0.201612903225806 0.409722222222222
## 3 Englishlan    low.friends 24   0.104838709677419 0.583333333333333
## 4   numberim   high.friends 27   0.037162162162162 0.397435897435897
## 5     Remain   high.friends 28  0.0150602409638554               0.4
## 6 Englishlan   high.friends 28  -0.036144578313253               0.6
## 7   numberim middle.friends 29  0.0360778443113772           0.58125
## 8     Remain middle.friends 30   0.152027027027027 0.518292682926829
## 9 Englishlan middle.friends 30 -0.0337837837837839 0.670731707317073
##                x.sd      quantile.lower     quantile.upper
## 1 0.281054437144578  0.0305705567263788  0.254817571127503
## 2 0.364264316209387   0.047001008064516  0.356224798387097
## 3 0.363514589999985  0.0567396313364054  0.152937788018433
## 4  0.17879299922592 -0.0390465038902541  0.113370828214578
## 5  0.35716117169158 -0.0685767624067699 0.0986972443344808
## 6 0.383138188901696 -0.0882245499856569 0.0159353933591509
## 7 0.312083055184567 -0.0325475104984836  0.104703199121238
## 8 0.359644980772875  0.0910151222651222  0.213038931788932
## 9 0.351065451475237  -0.185573865720925  0.118006298153357
```

```
    final.mat.friends = final.mat ##for graph


#subsetting by income, not sure where/ if this is used in paper
#######income subsets
subsets = c('low.income','middle.income', 'high.income')
final.mat = matrix(nrow = 0, ncol = 8)
cat('beginning income subset inference \n')
```

```
## beginning income subset inference
```

```
for(subset in subsets){

    out.mat = matrix(nrow = length(repeats), ncol = 8)

    if(subset == 'low.income'){
        dat.subset = dat.all[dat.all$income.new < 105000,]
        }
    if(subset == 'middle.income'){
        dat.subset = dat.all[dat.all$income.new >= 105000 & dat.all$income.new <= 135000,]
        }
    if(subset == 'high.income'){
        dat.subset = dat.all[dat.all$income.new > 135000,]
```

```
        }

    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        dat.use = dat.subset[is.na(dat.subset$Y) == F,]

        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)

        station.treatment.table = table(dat.use$station,dat.use[,z.variable])
        no.control.stations = names(which(station.treatment.table[,1] == 0))
        no.treatment.stations = names(which(station.treatment.table[,2] == 0))
        dat.use = dat.use[!dat.use$station%in%c(no.control.stations,no.treatment.stations),]

        dat.use$station = factor(dat.use$station)
        dat.use$treated_unit = factor(dat.use$treated_unit)
        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treated_u
        probs = genprobexact(Z = dat.use[,z.variable], blockvar=dat.use$station, clustvar=dat.use$treate

        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)
        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = ate)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

        ##fill matrix
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = x.mean
        out.mat[j,6] = x.sd
        out.mat[j,7] = disp$quantile[1]
        out.mat[j,8] = disp$quantile[2]

        }
    final.mat = rbind(final.mat,out.mat)
    }
final.mat = as.data.frame(final.mat)
colnames(final.mat) = c('variable','subset','N','ate','x.mean','x.sd','quantile.lower','quantile.upper')
print(final.mat)
```

```
##     variable        subset  N                 ate            x.mean
## 1   numberim    low.income 25               0.125               0.5
## 2     Remain    low.income 26   0.236486486486486         0.4296875
## 3 Englishlan    low.income 26 -0.0202702702702702         0.5859375
## 4   numberim middle.income 50  0.0906625272925765 0.491228070175439
## 5     Remain middle.income 53  0.0364309603440039 0.413793103448276
```

```
## 6 Englishlan middle.income 53  0.0816411849020544 0.581896551724138
## 7   numberim   high.income 14  0.0799769850402762 0.472222222222222
## 8     Remain   high.income 15   0.101993865030675 0.508928571428571
## 9 Englishlan   high.income 15 -0.0789877300613496 0.732142857142857
##                x.sd       quantile.lower     quantile.upper
## 1 0.258198889747161 -0.00116586538461547  0.251165865384615
## 2 0.343337362305413   0.0676691973566972  0.405303775616275
## 3 0.351318799943226    -0.15800778050778   0.11746723996724
## 4 0.294915666973741   0.0397412997630163  0.141583754822137
## 5 0.382029793943319  -0.0271839226334634  0.100045843321471
## 6 0.387264164325021   0.0226006004154981  0.140681769388611
## 7 0.243505383502854   0.0461570411392405  0.113796928941312
## 8 0.343471809029334   0.0468060106432423  0.157181719418107
## 9 0.311274050302197   -0.202502643381669 0.0445271832589701
```

```
final.mat.income = final.mat  ##for later
```

```
 ##primetest.r
 ###use randomization inference to look for changes in responses induced by experiment in subsets by co
 ####RdE September 2012


 #####recodes
 ####race variable
##primetest.r
###use randomization inference to look for changes in responses induced by experiment in subsets by cova
####RdE September 2012

#presumably this is one of the randomization parts
#not clear on what this is trying to demonstrate, although the nonsignficant p-values indicate
#that this is possibly the second randomization test (for missingness)?

#####recodes
####race variable
dat.all.prime$non.white = ifelse(dat.all.prime$race_4 == 1, 1,0)
dat.all.prime$non.white[is.na(dat.all.prime$non.white)==T] = 0


dat.all.prime$residency.new[dat.all.prime$residency == 1] = 1
dat.all.prime$residency.new[dat.all.prime$residency == 2] = 3.5
dat.all.prime$residency.new[dat.all.prime$residency == 3] = 7.5
dat.all.prime$residency.new[dat.all.prime$residency == 4] = 12.5
dat.all.prime$residency.new[dat.all.prime$residency == 5] = mean(dat.all.prime$age, na.rm = T)-15
dat.all.prime$residency.new[dat.all.prime$residency == 6] = mean(dat.all.prime$age, na.rm = T)

#####recodes
###English language is reverse coded from other variables:
dat.all.prime$Englishlan.x = recode(dat.all.prime$Englishlan.x, "5=1; 4=2; 3=3; 2=4; 1=5")
dat.all.prime$Englishlan.y = recode(dat.all.prime$Englishlan.y, "5=1; 4=2; 2=4; 1=5")


###gender recode
dat.all.prime$male = ifelse(dat.all.prime$gender == 1, 1, 0)
```

```
###inference
repeats = c("numberim","Remain","Englishlan")

x.names = paste(repeats,".x",sep="")
y.names = paste(repeats,".y",sep="")

covariates = c('line')

final.mat = matrix(nrow = 0, ncol = 8)

subsets = c('all.prime')

cat('beginning inference \n')
```

```
## beginning inference
```

```
for(subset in subsets){ ##b.only, complier, and non-compler subsets
    out.mat = matrix(nrow = length(repeats), ncol = 8)

    dat.subset = dat.all.prime

    z.variable = 'treatment'

    for(j in 1:length(repeats)){
        dat.subset$x.new = (as.numeric(dat.subset[,x.names[j]])-1)/4  ##rescale x to 0-1
        dat.subset$y.new = (as.numeric(dat.subset[,y.names[j]])-1)/4  ##rescale y to 0-1
        dat.subset$Y = dat.subset$y.new - dat.subset$x.new

        dat.use = dat.subset[is.na(dat.subset$Y) == F,]

        x.sd = sd(dat.use$x.new,na.rm = T)
        x.mean = mean(dat.use$x.new,na.rm = T)

        Xs = data.matrix(dat.use[,covariates])

        perms <- genperms(Z = dat.use[,z.variable])
        probs = genprobexact(Z = dat.use[,z.variable])
        ate = estate(Y = dat.use$Y, Z = dat.use[,z.variable], X = Xs, prob = probs)

        Ys = genouts(Y = dat.use$Y, Z = dat.use[,z.variable], ate = 0)
        distout <- gendist(Ys,perms, prob=probs)
        disp =  dispdist(distout, ate = ate, display.plot = F)

        ##fill matrix
        out.mat[j,1] = repeats[j]
        out.mat[j,2] = subset
        out.mat[j,3] = nrow(dat.use)
        out.mat[j,4] = ate
        out.mat[j,5] = disp$greater.p.value
        out.mat[j,6] = disp$lesser.p.value
        out.mat[j,7] = x.sd
        out.mat[j,8] = x.mean
        }
        final.mat = rbind(final.mat,out.mat)
```

```
        }
```

```
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least 4537567650 to perform exact estimation.
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least 9075135300 to perform exact estimation.
## Too many permutations to use exact method.
## Defaulting to approximate method.
## Increase maxiter to at least 9075135300 to perform exact estimation.
```

```
    final.mat = as.data.frame(final.mat)
    colnames(final.mat) = c('variable','subset','N','ate','greater.p.value','lesser.p.value','x.sd','x.m
    print(final.mat)
```

```
##     variable    subset  N                ate greater.p.value
## 1   numberim all.prime 35 0.0169249529219276          0.3082
## 2     Remain all.prime 36 0.0232323232323228          0.4421
## 3 Englishlan all.prime 36 0.0282828282828281           0.288
##   lesser.p.value              x.sd            x.mean
## 1         0.6918 0.199263349246521              0.55
## 2         0.5579 0.288932765288298 0.395833333333333
## 3          0.712   0.3034196632776 0.722222222222222
```

```
    final.mat.prime = final.mat ##mat for creating output later
```

```
#output_create.r

# ####create output
 output.vars = c('numberim','Remain','Englishlan')
 var.names = c('Number of immigrants be increased?','Children of undocumented be allowed to stay?','Engl


 ##main results
 ##cumalative results output
 final.mat.use = rbind(final.mat.main,final.mat.prime)

 final.mat.use$greater.p.value = as.numeric(as.character(final.mat.use$greater.p.value)); final.mat.use$
 final.mat.use$p.value = final.mat.use$greater.p.value

 final.mat.redact = final.mat.use[,c('variable','subset','ate','p.value','x.mean','x.sd','N')]
 final.mat.redact[,c('ate','p.value','x.mean','x.sd')] = round(final.mat.redact[,c('ate','p.value','x.me

 final.mat.redact$ate.new = paste(final.mat.redact$ate,' (',final.mat.redact$p.value,')',sep='')
 final.mat.redact$x.mean.new = paste(final.mat.redact$x.mean,' (',final.mat.redact$x.sd,')',sep='')

 out.mat.a = final.mat.redact[final.mat.redact$subset == 'all'&final.mat.redact$variable %in% output.var

 out.mat.a = final.mat.redact[final.mat.redact$subset == 'all'&final.mat.redact$variable %in% output.var
 out.mat.c = final.mat.redact[final.mat.redact$subset == 'no.car'&final.mat.redact$variable %in% output.
 out.mat.x = final.mat.redact[final.mat.redact$subset == 'all'&final.mat.redact$variable %in% output.var
 Ns = c('N',max(final.mat.redact$N[final.mat.redact$subset=='all']),
    max(final.mat.redact$N[final.mat.redact$subset=='no.car']),
    max(final.mat.redact$N[final.mat.redact$subset=='all'])
```

```
   )

 h1 = c('',paste('(',1:3,')',sep = ''))
 h2 = c('','all respondents','waits on platform','all respondents')
 h3 = c('question','ATE (p)','CATE (p)','T1 levels (sd)')
 hs = rbind(h1,h2,h3)
 row.names(hs) = NULL

 out.mat = cbind(out.mat.a,cbind(out.mat.c,out.mat.x))
 out.mat = cbind(var.names,out.mat)
 out.mat = rbind(out.mat,Ns)

 out.mat = rbind(hs,out.mat)

 out.table = xtable(out.mat, digits = 3
   )
 print(out.table,file = 'cum_results_wide.tex',
     floating = FALSE,
     include.rownames = FALSE,
     include.colnames = FALSE)

#not clear what "prime" means or in general how/ why the datasets are split
#this is a good topic to explore for next week

 ##prime results
 out.mat.p = final.mat.redact[final.mat.redact$subset == 'all.prime'&final.mat.redact$variable %in% outp
 Ns = c('N',max(final.mat.redact$N[final.mat.redact$subset=='all.prime']),
    max(final.mat.redact$N[final.mat.redact$subset=='all'])
    )

 h1 = c('',paste('(',1:2,')',sep = ''))
 h2 = c('','prime experiment','all respondents')
 h3 = c('question','ATE (p)','ATE (p)')
 hs = rbind(h1,h2,h3)
 row.names(hs) = NULL

 out.mat = cbind(out.mat.p,out.mat.a)
 out.mat = cbind(var.names,out.mat)
 out.mat = rbind(out.mat,Ns)
 out.mat = rbind(hs,out.mat)
 out.table = xtable(out.mat, digits = 3
   )
 print(out.table,file = 'prime_results_wide.tex',
     floating = FALSE,
     include.rownames = FALSE,
     include.colnames = FALSE)
```

```
##outgraphic_single.r
###create ouptput plots
####RdE November 2012

# ####create output
output.vars = c('numberim','Remain','Englishlan')
var.names = c('Number of immigrants be increased?','Children of undocumented be allowed to stay?','Engli
```

```
##graph presets
os = .4
line.os = .015
y.point = .75
ylims = c(0,1.1)
xlims = c(-.35,.35)
points.cex = 4
lab.cex = 1.5
line.lwd = 4.5
axis.cex = 1.25

colors = brewer.pal(3,'Paired')[1:2] ##colors for pairs used in plots below

####dose graph
pdf('dose_combined.pdf',
    width = 6.5, height = 4
    )

par(mfrow = c(3,1))
par(mar = c(5,0,1,0))
par(bty = 'n')


##dose response graph
#does this make Figure 2?
out.mat = final.mat.dose[,c('variable','subset','ate','quantile.lower','quantile.upper')]
out.mat$ate = as.numeric(as.character(out.mat$ate))
out.mat$quantile.lower = as.numeric(as.character(out.mat$quantile.lower))
out.mat$quantile.upper = as.numeric(as.character(out.mat$quantile.upper))

out.mat.ta = out.mat[out.mat$subset == 'ta'&out.mat$variable %in% output.vars,]
out.mat.tb = out.mat[out.mat$subset == 'tb'&out.mat$variable %in% output.vars,]

for(i in 1:length(var.names)){
    plot(x  = out.mat.ta$ate[i], y = y.point,
        xlim = xlims,
        ylim = ylims,
        ylab = '',
        xlab = var.names[i],
        yaxt = 'n',
        type = 'n',
        cex.lab = lab.cex,
        cex.axis = axis.cex)
    lines(x = c(out.mat.ta$quantile.lower[i],out.mat.ta$ate[i]-line.os),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)
    lines(x = c(out.mat.ta$ate[i]+line.os,out.mat.ta$quantile.upper[i]),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)
```

```
    lines(x = c(out.mat.tb$quantile.lower[i],out.mat.tb$ate[i]-line.os),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)
    lines(x = c(out.mat.tb$ate[i]+line.os,out.mat.tb$quantile.upper[i]),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)

    points(x  = out.mat.ta$ate[i], y = y.point,
        pch = 19,
        cex = points.cex,
        col = colors[1])
    points(x  = out.mat.tb$ate[i], y = y.point - os,
        pch = 1,
        cex = points.cex,
        col = colors[2])
            }
dev.off()
```

```
## pdf
##   2
```

```
##############################################
###graph presets

#this section and the one following format graphs for the supplemental inforamation portion
#(subsetting by ideology, "friends", and income, and testing facial characteristics)

os = .35
line.os = .01
y.point = .5
ylims = c(0,1.1)
xlims = c(-.35,.35)
points.cex = 1.25
lab.cex = 1.5
line.lwd = 2.5
axis.cex = 1

mean.label.x.os = .04
mean.label.y.os = .12
x.lim.size = 1
colors = brewer.pal(9,'Purples')[c(5,7,9)] ##colors.triple used in plots below

pdf('ideology_combined.pdf',
    width = 6.5, height = 4
    )
par(mfrow = c(3,1))
par(mar = c(5,0,1,0))
par(bty = 'n')

##ideology subset graph
```

```
out.mat = final.mat.ideology[,c('variable','subset','ate','x.mean','x.sd','quantile.lower','quantile.up
out.mat$ate = as.numeric(as.character(out.mat$ate));out.mat$quantile.lower = as.numeric(as.character(ou
out.mat[,c('x.mean','x.sd')] = round(out.mat[,c('x.mean','x.sd')],2)
out.mat$new.x.mean =  paste(out.mat$x.mean,' (',out.mat$x.sd,')',sep='')
out.mat.liberal = out.mat[out.mat$subset == 'liberals.only'&out.mat$variable %in% output.vars,]
out.mat.conservative = out.mat[out.mat$subset == 'conservatives.only'&out.mat$variable %in% output.vars
out.mat.moderate = out.mat[out.mat$subset == 'moderates.only'&out.mat$variable %in% output.vars,]

for(i in 1:length(var.names)){
    plot(x  = out.mat.moderate$ate[i], y = y.point,
        xlim = xlims,
        ylim = ylims,
        ylab = '',
        xlab = var.names[i],
        yaxt = 'n',
        type = 'n',
        cex.lab = lab.cex,
        cex.axis = axis.cex)

        lines(x = c(out.mat.liberal$ate[i]+line.os,out.mat.liberal$quantile.upper[i]),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)
        lines(x = c(out.mat.liberal$quantile.lower[i],out.mat.liberal$ate[i]-line.os),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)

        lines(x = c(out.mat.moderate$ate[i]+line.os,out.mat.moderate$quantile.upper[i]),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)
        lines(x = c(out.mat.moderate$quantile.lower[i],out.mat.moderate$ate[i]-line.os),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)

        lines(x = c(out.mat.conservative$ate[i]+line.os,out.mat.conservative$quantile.upper[i]),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)
        lines(x = c(out.mat.conservative$quantile.lower[i],out.mat.conservative$ate[i]-line.os),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)

        ###x means
```

```
        text(x  = out.mat.liberal$ate[i]+mean.label.x.os, y = y.point+os+mean.label.y.os,
            labels = out.mat.liberal$new.x.mean[i],
            cex = x.lim.size,
            col = colors[1])
        text(x  = out.mat.moderate$ate[i]+mean.label.x.os, y = y.point+mean.label.y.os,
            labels = out.mat.moderate$new.x.mean[i],
            cex = x.lim.size,
            col = colors[2])
        text(x  = out.mat.conservative$ate[i]+mean.label.x.os, y = y.point-os+mean.label.y.os,
            labels = out.mat.conservative$new.x.mean[i],
            cex = x.lim.size,
            col = colors[3])

    ###labels
    points(x  = out.mat.liberal$ate[i], y = y.point+os,
        pch = "L",
        cex = points.cex,
        col = colors[1])
    points(x  = out.mat.moderate$ate[i], y = y.point,
        pch = "M",
        cex = points.cex,
        col = colors[2])
    points(x  = out.mat.conservative$ate[i], y = y.point-os,
        pch = "C",
        cex = points.cex,
        col = colors[3])

    }
dev.off()
```

```
## pdf
##   2
```

```
############################################################
colors = brewer.pal(9,'Greens')[c(5,7,9)] ##colors.triple used in plots below

pdf('friends_combined.pdf',
    width = 6.5, height = 3.75
    )
par(mfrow = c(3,1))
par(mar = c(5,0,1,0))
par(bty = 'n')


###friends graph
out.mat = final.mat.friends[,c('variable','subset','ate','x.mean','x.sd','quantile.lower','quantile.upp
out.mat$ate = as.numeric(as.character(out.mat$ate));out.mat$quantile.lower = as.numeric(as.character(ou
out.mat[,c('x.mean','x.sd')] = round(out.mat[,c('x.mean','x.sd')],2)
out.mat$new.x.mean =  paste(out.mat$x.mean,' (',out.mat$x.sd,')',sep='')

out.mat.low.friends = out.mat[out.mat$subset == 'low.friends'&out.mat$variable %in% output.vars,]
out.mat.middle.friends = out.mat[out.mat$subset == 'middle.friends'&out.mat$variable %in% output.vars,]
out.mat.high.friends = out.mat[out.mat$subset == 'high.friends'&out.mat$variable %in% output.vars,]
```

```
for(i in 1:length(var.names)){
    plot(x  = out.mat.middle.friends$ate[i], y = y.point,
        xlim = xlims,
        ylim = ylims,
        ylab = '',
        xlab = var.names[i],
        yaxt = 'n',
        type = 'n',
        cex.lab = lab.cex,
        cex.axis = axis.cex)

        lines(x = c(out.mat.low.friends$ate[i]+line.os,out.mat.low.friends$quantile.upper[i]),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)
        lines(x = c(out.mat.low.friends$quantile.lower[i],out.mat.low.friends$ate[i]-line.os),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)

        lines(x = c(out.mat.middle.friends$ate[i]+line.os,out.mat.middle.friends$quantile.upper[i]),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)
        lines(x = c(out.mat.middle.friends$quantile.lower[i],out.mat.middle.friends$ate[i]-line.os),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)

        lines(x = c(out.mat.high.friends$ate[i]+line.os,out.mat.high.friends$quantile.upper[i]),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)
        lines(x = c(out.mat.high.friends$quantile.lower[i],out.mat.high.friends$ate[i]-line.os),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)

        ###x means
        text(x  = out.mat.low.friends$ate[i]+mean.label.x.os, y = y.point+os+mean.label.y.os,
            labels = out.mat.low.friends$new.x.mean[i],
            cex = x.lim.size,
            col = colors[1])
        text(x  = out.mat.middle.friends$ate[i]+mean.label.x.os, y = y.point+mean.label.y.os,
            labels = out.mat.middle.friends$new.x.mean[i],
            cex = x.lim.size,
            col = colors[2])
```

```
        text(x  = out.mat.high.friends$ate[i]+mean.label.x.os, y = y.point-os+mean.label.y.os,
            labels = out.mat.high.friends$new.x.mean[i],
            cex = x.lim.size,
            col = colors[3])

    ###labels
    points(x  = out.mat.low.friends$ate[i], y = y.point+os,
        pch = "L",
        cex = points.cex,
        col = colors[1])
    points(x  = out.mat.middle.friends$ate[i], y = y.point,
        pch = "M",
        cex = points.cex,
        col = colors[2])
    points(x  = out.mat.high.friends$ate[i], y = y.point-os,
        pch = "H",
        cex = points.cex,
        col = colors[3])
    }
dev.off()
```

```
## pdf
##   2
```

```
################################################
colors = brewer.pal(9,'OrRd')[c(5,7,9)] ##colors.triple used in plots below

pdf('income_combined.pdf',
    width = 6.5, height = 3.75
    )
par(mfrow = c(3,1))
par(mar = c(5,0,1,0))
par(bty = 'n')


############income graph
out.mat = final.mat.income[,c('variable','subset','ate','x.mean','x.sd','quantile.lower','quantile.upper
out.mat$ate = as.numeric(as.character(out.mat$ate));out.mat$quantile.lower = as.numeric(as.character(ou
out.mat[,c('x.mean','x.sd')] = round(out.mat[,c('x.mean','x.sd')],2)
out.mat$new.x.mean =  paste(out.mat$x.mean,' (',out.mat$x.sd,')',sep='')

out.mat.low.income = out.mat[out.mat$subset == 'low.income'&out.mat$variable %in% output.vars,]
out.mat.middle.income = out.mat[out.mat$subset == 'middle.income'&out.mat$variable %in% output.vars,]
out.mat.high.income = out.mat[out.mat$subset == 'high.income'&out.mat$variable %in% output.vars,]


for(i in 1:length(var.names)){
    plot(x  = out.mat.middle.income$ate[i], y = y.point,
        xlim = xlims,
        ylim = ylims,
        ylab = '',
        xlab = var.names[i],
        yaxt = 'n',
        type = 'n',
```

```
        cex.lab = lab.cex,
        cex.axis = axis.cex)

        lines(x = c(out.mat.low.income$ate[i]+line.os,out.mat.low.income$quantile.upper[i]),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)
        lines(x = c(out.mat.low.income$quantile.lower[i],out.mat.low.income$ate[i]-line.os),
            y = c(y.point+os,y.point+os),
            lty = 1,
            col = colors[1],
            lwd = line.lwd)

        lines(x = c(out.mat.middle.income$ate[i]+line.os,out.mat.middle.income$quantile.upper[i]),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)
        lines(x = c(out.mat.middle.income$quantile.lower[i],out.mat.middle.income$ate[i]-line.os),
            y = c(y.point,y.point),
            lty = 1,
            col = colors[2],
            lwd = line.lwd)

        lines(x = c(out.mat.high.income$ate[i]+line.os,out.mat.high.income$quantile.upper[i]),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)
        lines(x = c(out.mat.high.income$quantile.lower[i],out.mat.high.income$ate[i]-line.os),
            y = c(y.point-os,y.point-os),
            lty = 1,
            col = colors[3],
            lwd = line.lwd)

        ###x means
        text(x  = out.mat.low.income$ate[i]+mean.label.x.os, y = y.point+os+mean.label.y.os,
            labels = out.mat.low.income$new.x.mean[i],
            cex = x.lim.size,
            col = colors[1])
        text(x  = out.mat.middle.income$ate[i]+mean.label.x.os, y = y.point+mean.label.y.os,
            labels = out.mat.middle.income$new.x.mean[i],
            cex = x.lim.size,
            col = colors[2])
        text(x  = out.mat.high.income$ate[i]+mean.label.x.os, y = y.point-os+mean.label.y.os,
            labels = out.mat.high.income$new.x.mean[i],
            cex = x.lim.size,
            col = colors[3])

    ###labels
    points(x  = out.mat.low.income$ate[i], y = y.point+os,
        pch = "L",
```

```
        cex = points.cex,
        col = colors[1])
    points(x  = out.mat.middle.income$ate[i], y = y.point,
        pch = "M",
        cex = points.cex,
        col = colors[2])
    points(x  = out.mat.high.income$ate[i], y = y.point-os,
        pch = "H",
        cex = points.cex,
        col = colors[3])

    }
dev.off()
```

```
## pdf
##   2
```

```
###face_summary.r
###give summary statistics for each face in confederate sample and comparison sample, make comparisons
###RdE February 2013


##se function
stderr <- function(x) sqrt(var(x,na.rm=TRUE)/length(na.omit(x)))

out.list = list()
for(i in 1:3){
    if(i == 1){use.dat = conf.dat}
    if(i == 2){use.dat = hisp.dat}
    if(i == 3){use.dat = white.dat}


    vars.keep = c('race_1','his','_imm','_for','fri','app','IQ','intim','suc','wea','UE','_pro','ind','y


    out.mat = matrix(ncol = length(vars.keep), nrow = nrow(use.dat)*7) ##7 faces, so when data is stacke
    colnames(out.mat) = vars.keep

    for(var in vars.keep){
        use.cols = grep(var,colnames(use.dat), fixed = T)
        this.dat = use.dat[,use.cols]
        col.dat = as.vector(as.matrix(this.dat))
        out.mat[,var] = col.dat
        }
    ##recode 0/1
    out.mat[, c('_imm','_for','intim','fri','app','IQ','suc','wea','you','_pro','ind','UE','_com','unu'
    (out.mat[, c('_imm','_for','intim','fri','app','IQ','suc','wea','you','_pro','ind','UE','_com','unu

    if(i == 1){conf.out = out.mat}
    if(i == 2){hisp.out = out.mat}
    if(i == 3){white.out = out.mat}


    out.means = apply(out.mat,2,mean, na.rm = T)
```

```
    out.ses = apply(out.mat,2,stderr)

    out.final = rbind(out.means,out.ses)

    out.list[[i]] = out.final
    }


x.names = c('white','Hispanic','immigrant','foreign','friendly','approachable','intelligent','intimidati


ylims = c(.5,ncol(out.list[[1]])*2)+.5
ylims = c(.5,ncol(out.list[[1]])*2)+.5
xlims = c(-.1,1)
yrange = (1:ncol(out.list[[1]]))*2

plot.sets = list(c(1,2,3,4),
    c(5,6,7,8),
    c(9,10,11,12),
    c(13,14,15,16)
    )

pdf('face_ratings_2.pdf',
    width = 9, height = 6.95
    )
par(las = 1)
par(mfrow = c(2,2))
par(mar = c(2.5, .5, 1.5, .5))
par(bty = 'n')
for(h in 1:length(plot.sets)){
    use.set = plot.sets[[h]]
    use.names = x.names[use.set]
    plot(out.list[[1]][1,use.set],1:4,
        xlim = xlims,
        ylim = c(.5,4.5),
        type = 'n',
        yaxt = 'n',
        ylab = '',
        cex.lab = 1.75
        )

    for(i in 1:4){

        lab.place.mean = min(out.list[[1]][1,use.set][i],
                        out.list[[2]][1,use.set][i],
                        out.list[[3]][1,use.set][i]
                        )
        lab.place.se = min(out.list[[1]][2,use.set][i],
                        out.list[[2]][2,use.set][i],
                        out.list[[3]][2,use.set][i]
                        )
        text(x = lab.place.mean - (2*lab.place.se),
            y = i,
```

```
            labels = use.names[i],
            cex = 1.25,
            pos = 2
            )

        lines(x= c(out.list[[1]][1,use.set][i]+.015,out.list[[1]][1,use.set][i]+(1.96*out.list[[1]][2,us
            y = c(i,i),
            col = 'red',
            lty = 1)
        lines(x= c(out.list[[1]][1,use.set][i]-.015,out.list[[1]][1,use.set][i]-(1.96*out.list[[1]][2,us
            y = c(i,i),
            col = 'red',
            lty = 1)

        lines(x= c(out.list[[2]][1,use.set][i]+.015,out.list[[2]][1,use.set][i]+(1.96*out.list[[2]][2,us
            y = c(i+.25,i+.25),
            col = 'blue',
            lty = 1)
        lines(x= c(out.list[[2]][1,use.set][i]-.015,out.list[[2]][1,use.set][i]-(1.96*out.list[[2]][2,us
            y = c(i+.25,i+.25),
            col = 'blue',
            lty = 1)

        lines(x= c(out.list[[3]][1,use.set][i]+.015,out.list[[3]][1,use.set][i]+(1.96*out.list[[3]][2,us
            y = c(i-.25,i-.25),
            lty = 1)
        lines(x= c(out.list[[3]][1,use.set][i]-.015,out.list[[3]][1,use.set][i]-(1.96*out.list[[3]][2,us
            y = c(i-.25,i-.25),
            lty = 1)
            }
    text(out.list[[1]][1,use.set],1:4,
        'C',
        col = 'red',
        cex = 1)
    text(out.list[[2]][1,use.set],1:4+.25,
        'H',
        col = 'blue',
        cex = 1)
    text(out.list[[3]][1,use.set],1:4-.25,
        'A',
    cex = 1)

    }
dev.off()
```

```
## pdf
##   2
```

```
###analysis_master.r
###master train analysis script
###RdE October 2012

rm(list = ls())
```

```
###THE PACKAGES BELOW MAY NEED TO BE INSTALLED USING install.packages('x'), WHERE X IS THE PACKAGE NAME
#copied to top of rmd file

# library(ri)
# library(RItools)
# library(car)
# library(xtable)
# library(effects)
# library(RColorBrewer)
############################




# ##the following scripts will be executed
# source('balance_check.r') ##executes balance tests
# source('main_results.r')  ##produces ATE and car subanalysis
# source('subset_inference.r')  ##produces dose effect and subgroup effects in supporting material
# source('primetest.r') ##produces prime test in supporting material
# source('output_create.r')  ##creates tables
# source('outgraphic_single_pnas.r') ##produces graphics, expect face graphic
# source('face_summary_pnas.r') ##creates face graphic
```