Homework 2

Loren Collingwood
CSSS 529 Survey Sampling
Professor Handcock

June 22, 2009

Problem 2.1

Problem 2.1a

The advantages and disadvantages of the two approaches are many, but the decision should based first and foremost on how many employees work at the HMO. Based on the language of the question, there are probably a lot of employees, which is informative. Below I list the key advantages and disadvantages:

- Cost: Assuming the population size is large, the cost for a sample of 10% should be much lower than if the full population was interviewed. Thus, this is an advantage for the sample and disadvantage for the census.

- Design complications: Drawing the sample may involve knowing in advance strata the sample may be stratified on, such as physicians versus nurses, or department type within the HMO. Drawing a representative stratified sample can be complicated, so there is error involved with the sampling, but not with the census.

- Response rate issues: This is always a potential problem for surveys. Sampling can be advantageous here if we can stratify the sample on variables that may improve our estimation. That way we can ensure a certain number of respondents, for instance, from each HMO profession are sampled. This may not be the case in a census survey, where, for instance, doctors may be more likely to respond to the survey than nurses. If this is the case, our survey will misrepresent HMO staff opinion.

  But, if the response rates were very high for both surveys, then we will have greater confidence in the results of the census survey.

Problem 2.1b

Useful variables would include the branch or department of the HMO, region or other geographic entity of the branch or department, type of job, and because the survey involves emergency preparedness, mode of transportation of the respondents, and distance from their work, gender.

Problem 2.1c

The sample probability for physicians is simply $450/900 = 0.5$, and the probability for other is $450/9000 = 0.05$.

Problem 2.1d

The Horvitz-Thompson estimator is used to determine the unbiased estimates of the proportion of *physicians* and *other workers* who can make it into work after an earthquake.

$$\hat{\tau}_\pi = \sum_{i=1}^{v} \frac{y_i}{\pi_i}$$

Because 300 out of 450 respondents said they could make it into work, this gives us a population proportion of 0.67. For the other workers, the proportion is 150/450, or 0.33. For the physician stratum, multiplying this figure by the probability of selection (0.5) gives us a multiplier of 1.33. This figure is then summed 450 times, for a total of 600. Six hundred divided by 900 (the total number of physicians) gives a stratum proportion of .67. The same process yields a stratum proportion of 0.33 for the other workers. The total proportion of workers estimated to be able to make it into work, therefore is 3600/9900 or 0.36.

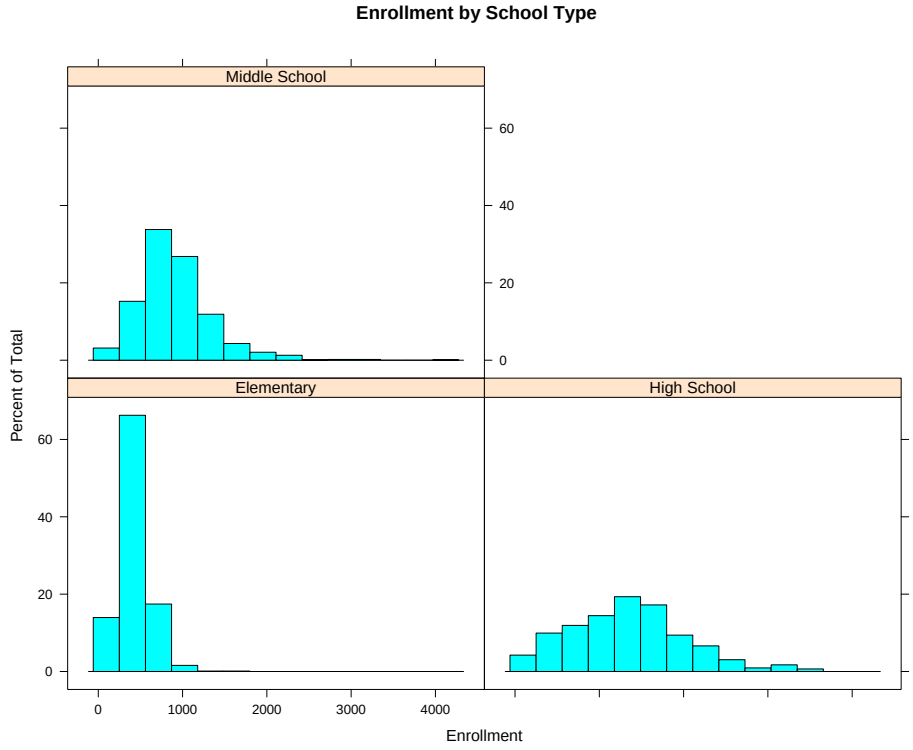
Problem 2.1e

Explaining to the managers that commissioned the study why simply adding up and dividing by the total is not the correct thing to do may be challenging. The reason is because the sample has been stratified by profession—physician versus other. An equal number of respondents were selected within each stratum, but because there are much fewer doctors overall, we must take into account their unequal probability of selection. According to this sample design, the probability of being sampled is .5 for physicians and .05 for staff, so the chances of a physician being sampled are much higher than that of the staff, so there is significant undersampling of the staff, therefore leading to misrepresentation of the answer choice for those who are able to commute to work if we do not include the probability weighting in our total estimation.

Problem 2.5

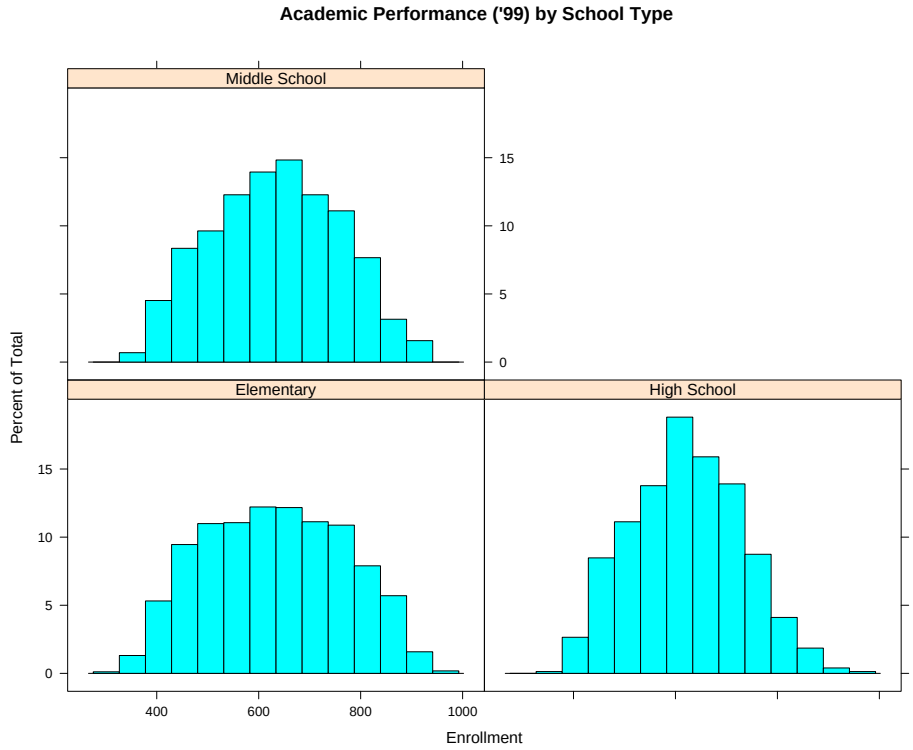
An examination of enrollment histograms by school type shows why school type is a critical variable to stratify on if we want to properly estimate the mean and total number of students.

Figure 1: *Histograms by school type reveal that there are key distributional differences by school type, specifically with regards to elementary schools, which tend to have a higher proportion of schools with fewer students.*



However, when examining histograms of academic performance by school type, no real differences emerge in terms of academic performance distribution by school type. To be sure, I correlated school type by api99; the correlation is basically zero. Thus, when stratifying the sample on school type, API is not affected since the distribution across schools varies little. I present a graph for the variable API99 by school type.

Figure 2: *Histograms by school type reveal that API does not vary by school type.*



A similar approach can be used to determine, a-priori, whether stratification on school type will affect other variables, but first I theorize as to why stratification of the variables mobility, emergency, meals, and pcttest will lead to great precision of estimating the mean value for the Academic Performance Index (API). Overall, these variables are likely correlated with the income of a community, and the income of a community is likely highly correlated with academic performance. The list of the variables is presented:

- Mobility = proportion of students who are new to the school
- Emer = proportion of teachers with only emergency qualifications
- Meals = proportion of students receiving subsidized meals

- Pcttest = percentage of students who took the API test

The above variables all correlate with academic performance. Indeed, correlations reveal that the more mobile the students are in a school, the lower the academic performance (correlation = -.20). The correlation jumps to -.51 for emergency, -.85 for meals, and .17 for students percentage of students who took the test.

I drew several random stratified samples for each of the variables above to determine whether accuracy for API means and totals improves. As you can see from the table below, there is indeed a noticeable improvement for standard errors, however, there is little improvement for sample means. This runs against theoretical prediction. Possibly my decision to stratify the variables along certain cut-points was incorrect, but not knowing the data completely it is hard to determine where the natural cut-points would be for stratification.

Table 1: *Academic Performance Sample Mean estimates by various stratifications*

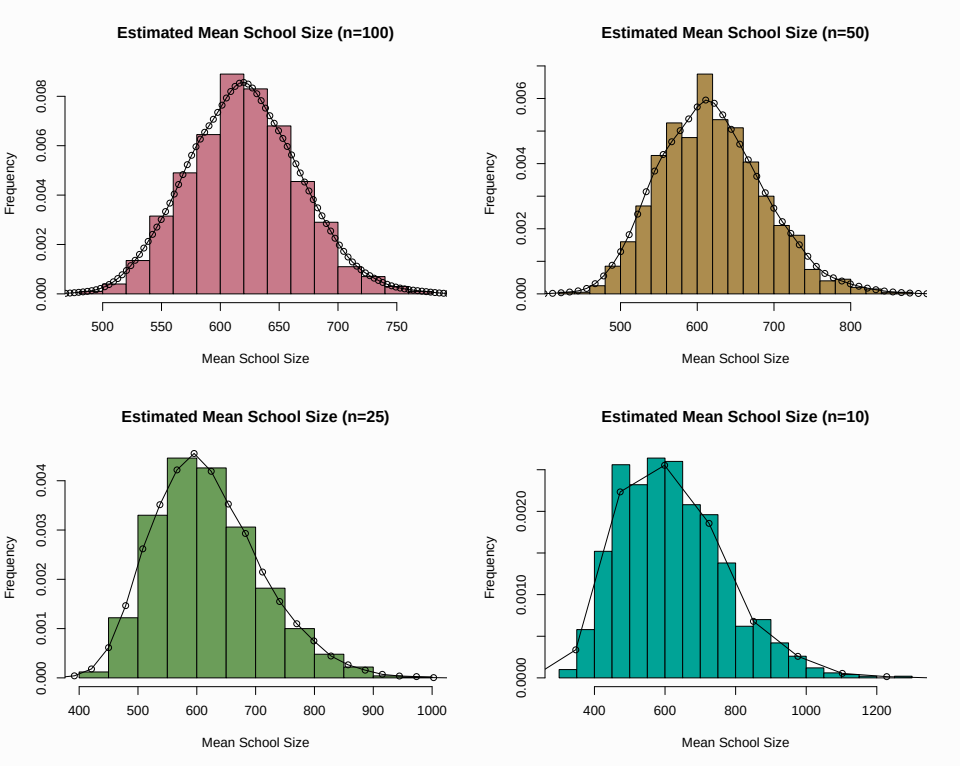
Sample	API99	Design - Actual	API00	Design - Actual
Actual Mean	631.9		664.7	
SRS Mean	630.57	-1.33	663.97	-0.73
SRS SE	13.31		13.3	
Strat School Mean	623.09	-8.81	653.43	-11.27
School SE	14.2		14	
Strat Mobility Mean	624.91	-6.99	663.57	-1.13
Mobility SE	12.4		12.4	
Strat Emer Mean	645.41	13.51	682.69	17.99
Emergency SE	10.3		9.95	
Strat Meals Mean	642.98	11.08	673.12	8.42
Meals SE	8.65		12.83	
Strat PCT Test Mean	645.55	13.65	675.57	10.87
PCT Test SE	12.95		12.43	

Note, because this question was ambiguously worded and somewhat misleading, I also stratified the sample normally—based on school type—and looked at the means and totals of emergency, meals, percent who take the test, and mobility. Here, for means I find that there is some precision increase for the variables emergency and meals, but no increase in terms of standard errors for mobility and pcttest. In general, I do not see much of a difference.

Problem 2.7

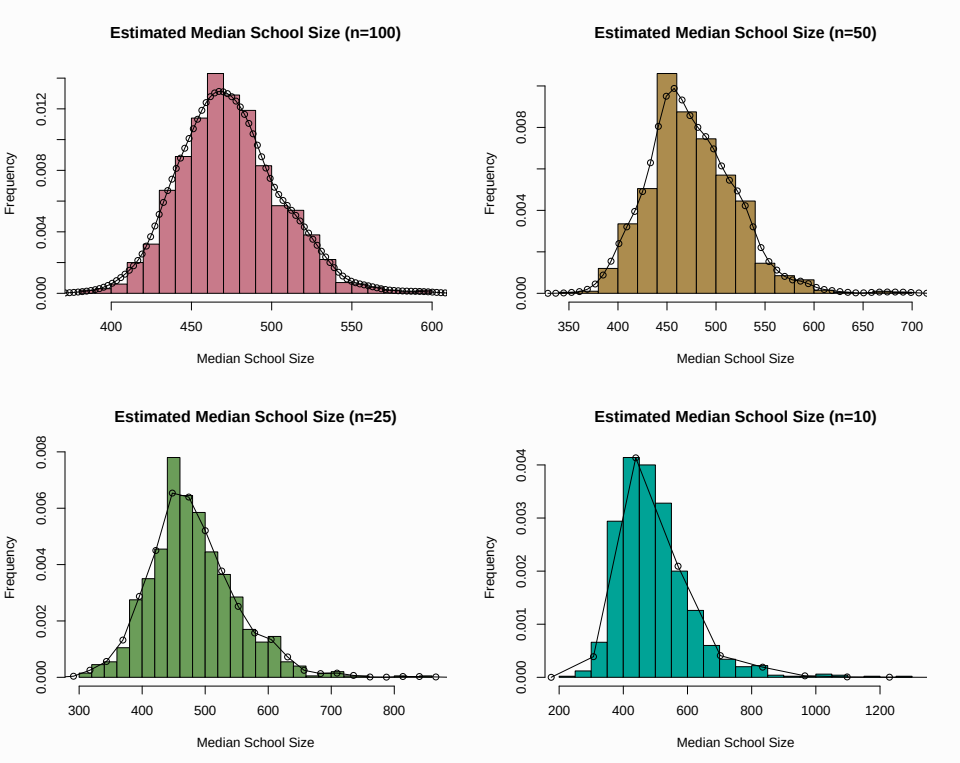
2.7a

Figure 3: *Simulation Results: As the sample size decreases, the distribution for the mean value of school size becomes less normally distributed*



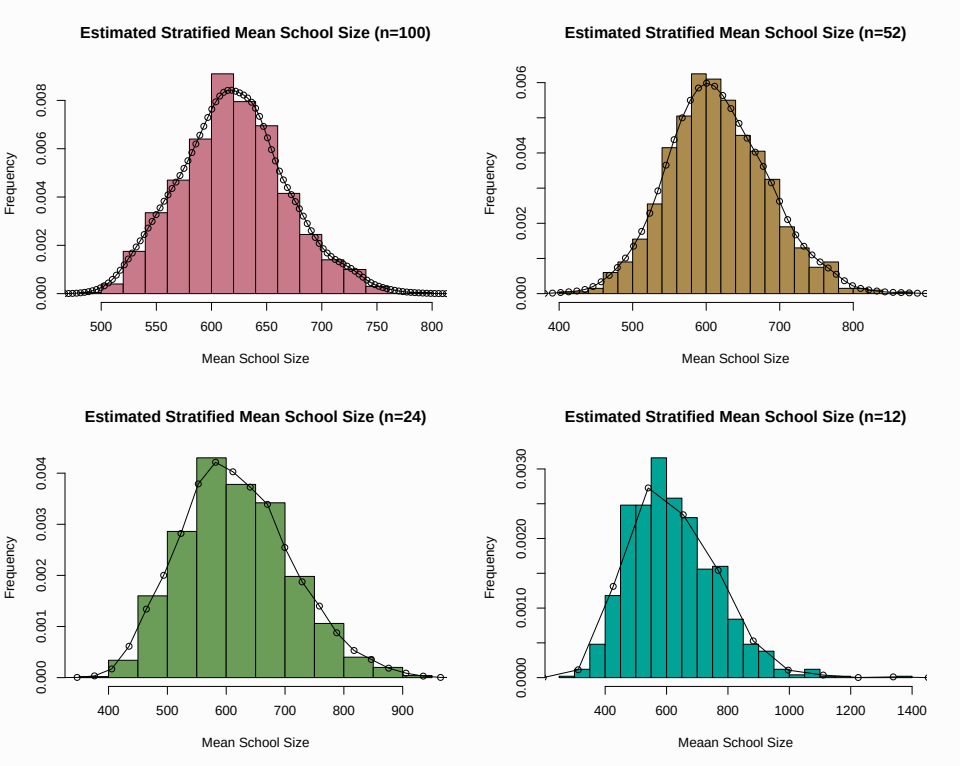
2.7b

Figure 4: *Simulation Results: As the sample size decreases, the distribution for the median value of school size becomes less normally distributed*



2.7c

Figure 5: *Simulation Results: Stratified Sample*



Problem 3.7

3.7a

Choosing a simple random sample from 10 counties I used the cluster function in R, simple random sampling without replacement. This method produces reasonably accurate results but with large variance. Indeed, I reran this sample a few times and the murder and burglary counts vary considerably depending on the counties selected.

Table 2: *Estimated number of murders and burglaries in WA State (2004) using simple random sampling by county*

Crime	Total	Standard Error
Actual Murder	189	
SRS Murder	191.1	70.26
Actual Burglary	59,985	
SRS Burglary	57,545	21,506

3.7b

The sample was divided such that all units within King county were selected as well as five other counties. This was done by dividing the data into two universes: one for King, and one for all others. King was assigned a 100% probability weight, whereas others were assigned a much smaller weight of .1356 (taken from SRSWOR). This produced significantly worse results for predicting statewide total murders and burglaries than the previous sample

At least in part because King is the largest county in the state, this sampling method and using the cluster command for the for two separate sub-setted samples. The probability option in R gives probability weights for each row in the data that allows us to calculate the probability of selection based on the design.

Table 3: *Estimated number of murders and burglaries in WA State, full proportional sampling for King County and five other counties.*

Crime	Total	Standard Error
Actual Murder	189	
SRS II Murder	87.4	50.9
Actual Burglary	59,985	
SRS II Burglary	30,901	15,156

3.7c

This sampling approach significantly drops our estimation abilities. Just 19 murders and 4927 burglaries are estimated using this approach. While these number are very low, we would expect them to be at least a little higher. We are limiting our generalizable abilities when we cluster the sample this way, and evidently are not bringing in enough data points.

Crime	Total	SE
Murder Total	19	144.8
Burglary Total	4927	2996.33

Table 4: Murder and Burglary totals and and standard errors. This clustering approach appears to be significantly less accurate.

3.7d

Sampling via probability proportionate to size improves the estimation for total number of murders but not the total number of burglaries. In fact, my results indicate that burglaries are drastically over-estimated.

Crime	Total	SE
Murder Total	227.18	62.81
Burglary Total	108,890.44	18,904.86

Table 5: Murder, burglary totals and standard errors. This clustering approach improves estimates for murder but clearly not burglary.

3.7e

Taking a simple random sample of counties and including all the precincts within these counties is a similar approach to question 3.7a. The difference is that I can

choose the sample size, which, I arbitrarily decide as 20. In theory, this should vastly improve our total estimates because there n is larger. And it does improve our estimates to some degree as can be witnessed in the table below. Nevertheless, the estimates are not as spectacular as we would hope; which speaks to the downside of the clustering method.

Crime	Total	SE
Murder Total	253.50	82.31
Burglary Total	80,907.45	26,160.95

Table 6: Murder, burglary totals and standard errors of 20 randomly selected counties.

Ideally, we would draw a probability proportionate to size sample based on Agency population size. This is the approach in the above question, however the sample size was not large enough to generate extremely accurate estimates.

```
#########################################################
#####           HOMEWORK 2 SURVEY SAMPLING          ####
#####                                               ####
#########################################################

###libraries
library(survey)
library(sampling)
library(foreign)
library(lattice)
library(MASS)
library(car)
library(boot)
library(RColorBrewer)
library(xtable)
library(quantreg)

###########READING IN DATA#################33
```

```
setwd ("C:/Users/Loren/Documents/UW courses/Spring 2009/CSSS 529/hm1")
calapi <- read.dta("calapi.dta")

############################################################
#                          PROBLEM 2.5                     #
############################################################

mean(calapi$api99)
svymean(~api00, srs_design, na.rm=T)
svymean(~api99, srs_design, na.rm=T)

############################################################
#           RECODING stype into numeric form               #
############################################################

calapi$stype2[calapi$stype == "E"] <-1
calapi$stype2[calapi$stype == "H"] <-2
calapi$stype2[calapi$stype == "M"] <-3

calapi$stype3[calapi$stype == "E"] <-"Elementary"
calapi$stype3[calapi$stype == "H"] <-"High School"
calapi$stype3[calapi$stype == "M"] <-"Middle School"

############################################################
#                      LATTICE GRAPHS                      #
############################################################

############################################################
#           ENROLLMENT BY SCHOOL TYPE, GRAPHING            #
############################################################
pdf(file="enrollstype.pdf",height=8, width=10)
histogram(~enroll | stype3, data = calapi,
            main = "Enrollment by School Type",
```

```
            xlab = "Enrollment")
dev.off()

############################################################
#ACADEMIC PERFORMANCE BY SCHOOL TYPE, GRAPHING
############################################################

pdf(file = "apistype.pdf", height=8, width=10)
histogram(~api99 | stype3, data = calapi,
            main = "Academic Performance ('99) by School Type",
            xlab = "Enrollment")
dev.off()

############################################################
##              CORRELATING VARIABLES                     ##
##  meals, pcttest, emer, mobility                        ##
##  Some of these var's have NA's so need to construct    ##
##  new data frames.                                      ##
############################################################

cor(calapi$stype2, calapi$api99)
cor(calapi$meals, calapi$api99)
cal.omit <- subset(calapi, select = c(snum, stype2, api99, api00,
enroll, meals, emer, pcttest, mobility))
cal.omit2 = na.omit(cal.omit)
cor(cal.omit2$emer, cal.omit2$api99)
cor(cal.omit2$pcttest, cal.omit2$api99)
cor(cal.omit2$mobility, cal.omit2$api99)
sum(calapi$api99, na.rm=T)
summary(calapi$api99)
summary(calapi$api00)

############################################################
#               STRATIFICATION OF VARIABLES                #
```

```
############################################################
#STRATIFICATION FUNCTION

stratsample<-function(strata, counts){
   strata<-as.character(strata)
   n<-length(strata)
   rval <- integer(sum(counts))
   allrows<-1:n
   j<-0
   for(i in 1:length(counts))    {
       thisstrat<-names(counts)[i]
      rval[j+(1:counts[i])]<-sample
      (allrows[strata==thisstrat],counts[i])
      j<-j+counts[i]
   }
   rval
   }

############################################################
#           CREATING SCHOOL TYPE STRATIFIED DATA FRAME      #
#FROM HOMEWORK 1                                            #
############################################################

strat_rows <- stratsample(calapi$stype, counts=c(E=50, M=25,H=25))
strat_data <- calapi[strat_rows,]
elem <- subset(strat_data, stype=="E")
elem$npopsize <- rep(4421, 50)
mid <- subset(strat_data, stype == "M")
mid$npopsize <- rep(1018, 25)
hs <- subset(strat_data, stype=="H")
hs$npopsize <- rep(755, 25)
fpcstrat <- rbind(data.frame(elem), data.frame(mid), data.frame(hs))
dim(fpcstrat) # all good
####################Generating the design#####################
```

```
strat_design <- svydesign(id=~snum, strata=~stype,
fpc=~npopsize, data=fpcstrat)
strat_design
svytotal(~enroll, strat_design, na.rm=T)
svymean(~enroll, strat_design, na.rm=T)
svytotal(~api00, strat_design, na.rm=T)
svymean(~api99, strat_design, na.rm=T)
svymean(~api00, strat_design, na.rm=T)
##############################################################
#                "MOBILITY" STRATIFIED DATA FRAME            #
##############################################################
cal.omit <- subset(calapi, select = c(snum, stype2, api99, api00,
enroll, meals, emer, pcttest, mobility))
cal.omit2 = na.omit(cal.omit)
summary(calapi$mobility)
hist(calapi$mobility)
calapi$mob <- cut(calapi$mobility, c(0,10,20,100), include.lowest = T,
right=F, labels=c('low','mid','high'))  ##USED FOR STRATIFICATION
histogram(~api99 | mob2, data = calapi)
table(calapi$mob2)

#####CREATING ACTUAL STRATA
strat_rowsmob <- stratsample(calapi$mob,
counts=c(low=25, mid=50, high=25))
strat_datamob <- calapi[strat_rowsmob,]
low <- subset(strat_datamob, mob=="low")
low$mpopsize <- rep(1102, 25)
mid <- subset(strat_datamob, mob == "mid")
mid$mpopsize <- rep(3315, 50)
high <- subset(strat_datamob, mob=="high")
high$mpopsize <- rep(1773, 25)
strat_mob <- rbind(data.frame(low), data.frame(mid), data.frame(high))
dim(strat_mob) # all good, bitches
####################Generating the design#####################
```

```
strat_designmob <- svydesign(id=~snum, strata=~mob,
fpc=~mpopsize, data=strat_mob)
strat_designmob
svytotal(~api99, strat_designmob, na.rm=T)
svymean(~api99, strat_designmob, na.rm=T)
svymean(~api00, strat_designmob, na.rm=T)

par(mfrow=c(1,2))
svyhist(~api00, strat_designmob)
hist(strat_mob$api00)

#################################################################
#                    EMERGENCY STRATIFICATION                   *
#################################################################

cal.omit2$em <- cut(cal.omit2$emer, quantile(cal.omit2$emer, (0:3)/3),
include.lowest=T, right=T, labels=c('low','mid','high'))
table(cal.omit2$em)

#####CREATING ACTUAL STRATA
strat_rowsem <- stratsample(cal.omit2$em,
counts=c(low=34, mid=34, high=32))
strat_dataem <- cal.omit2[strat_rowsem,]
low <- subset(strat_dataem, em=="low")
low$epopsize <- rep(2070, 34)
mid <- subset(strat_dataem, em == "mid")
mid$epopsize <- rep(2210, 34)
high <- subset(strat_dataem, em=="high")
high$epopsize <- rep(1871, 32)
strat_em <- rbind(data.frame(low), data.frame(mid), data.frame(high))
dim(strat_em) # all good

####################Generating the design#####################
strat_designem <- svydesign(id=~snum, strata=~em,
```

```
fpc=~epopsize, data=strat_em)
strat_designem
svytotal(~api99, strat_designem, na.rm=T)
svymean(~api99, strat_designem, na.rm=T)
svymean(~api00, strat_designem, na.rm=T)

#################################################################
#                    MEALS STRATIFICATION                       #
#################################################################

summary(calapi$meals)
hist(calapi$meals)
hist(cal.omit2$meals)
#RECODING, just trying out a few ways##
calapi$meal <- cut(calapi$meals, quantile(calapi$meals, (0:3)/3),
include.lowest=T, right=T, labels=c('low','mid','high'))
table(calapi$meal)

#####CREATING ACTUAL STRATA
strat_rowsmeal <- stratsample(calapi$meal, counts=c(low=34, mid=34,
high=32))
strat_datameal <- calapi[strat_rowsmeal,]
low <- subset(strat_datameal, meal=="low")
low$mlpopsize <- rep(2065, 34)
mid <- subset(strat_datameal, meal == "mid")
mid$mlpopsize <- rep(2085, 34)
high <- subset(strat_datameal, meal=="high")
high$mlpopsize <- rep(2044, 32)
strat_meal <- rbind(data.frame(low), data.frame(mid), data.frame(high)
dim(strat_meal) # all good

####################Generating the design#####################

strat_designmeal <- svydesign(id=~snum, strata=~meal,
```

```
fpc=~mlpopsize, data=strat_meal)
strat_designmeal
svytotal(~api99, strat_designmeal, na.rm=T)
svymean(~api99, strat_designmeal, na.rm=T)
svymean(~api00, strat_designmeal, na.rm=T)
par(mfrow=c(1,2))
svyhist(~api00, strat_designem)
hist(strat_mob$api00)

#################################################################
#                    Percent Test STRATIFICATION                *
#################################################################

summary(calapi$pcttest)
hist(calapi$pcttest)
hist(cal.omit2$pcttest)
#RECODING -- PCTTEST IS SO SKEWED IN DISTRIBUTION I'M DOING 95 AND OTH
cal.omit2$pct <- cut(cal.omit2$pcttest, c(0,98,100), include.lowest =
right=F, labels=c('low','high'))  ##USED FOR STRATIFICATION
table(cal.omit2$pct)
#####CREATING ACTUAL STRATA
strat_rowspct <- stratsample(cal.omit2$pct, counts=c(low=30, high=70))
strat_datapct <- cal.omit2[strat_rowspct,]
low <- subset(strat_datapct, pct=="low")
low$ppopsize <- rep(1016, 30)
high <- subset(strat_datapct, pct == "high")
high$ppopsize <- rep(5135, 70)
strat_pct <- rbind(data.frame(low), data.frame(high))
dim(strat_pct) # all good

####################Generating the design#####################

strat_designpct <- svydesign(id=~snum, strata=~pct,
fpc=~ppopsize, data=strat_pct)
```

```
strat_designpct
svytotal(~api99, strat_designpct, na.rm=T)
svymean(~api99, strat_designpct, na.rm=T)
svymean(~api00, strat_designpct, na.rm=T)

#####################################################
#           PROBLEM 2.7 A                           #
#####################################################

set.seed(1) # set the random number starting value
one.mean <- function(n){
        srs_rows<-sample(6194,n)
        mean(calapi$enroll[srs_rows],na.rm=TRUE)
    }

one.mean(100)
many.means100 <- replicate(1000, one.mean(100))
hist(many.means100)

many.means50 <- replicate(1000, one.mean(50))
hist(many.means50)

many.means25 <- replicate(1000, one.mean(25))
hist(many.means25)

many.means10 <- replicate(1000, one.mean(10))
hist(many.means10)

##################GRAPH GENERATION#############################

pdf(file="meanss.pdf", height=8, width=10)
par(mfrow=c(2,2), bg="gray99")
truehist(many.means100, main = "Estimated Mean School Size (n=100)",
xlab = "Mean School Size", ylab = "Frequency", col="#C87A8A")
```

```
lines(density(many.means100, width = "SJ-dpi", n=100), lty=1, type = "
truehist(many.means50, main = "Estimated Mean School Size (n=50)",
xlab = "Mean School Size", ylab = "Frequency", col="#AC8C4E")
lines(density(many.means50, width = "SJ-dpi", n=50), lty=1, type = "o"
truehist(many.means25, main = "Estimated Mean School Size (n=25)",
xlab = "Mean School Size", ylab = "Frequency", col="#6B9D59")
lines(density(many.means25, width = "SJ-dpi", n=25), lty=1, type = "o"
truehist(many.means10, main = "Estimated Mean School Size (n=10)",
xlab = "Mean School Size", ylab = "Frequency", col="#00A396")
lines(density(many.means10, width = "SJ-dpi", n=10), lty=1, type = "o"
dev.off()

##################################################
#            PROBLEM 2.7 B                       #
##################################################

one.med <- function(n){
        srs_rows<-sample(6194,n)
        median(calapi$enroll[srs_rows],na.rm=TRUE)
    }

many.med100 <- replicate(1000, one.med(100))
truehist(many.med100, main = "Estimated Median School Size (n=100)",
xlab = "Mean School Size", ylab = "Frequency", col="#C87A8A")
lines(density(many.med100, width = "SJ-dpi", n=100), lty=1, type = "o"
many.med50 <- replicate(1000, one.med(50))
truehist(many.med50, main = "Estimated Median School Size (n=50)",
xlab = "Mean School Size", ylab = "Frequency", col="#AC8C4E")
lines(density(many.med50, width = "SJ-dpi", n=50), lty=1, type = "o")
many.med25 <- replicate(1000, one.med(25))
truehist(many.med25, main = "Estimated Median School Size (n=25)",
xlab = "Mean School Size", ylab = "Frequency", col="#6B9D59")
lines(density(many.med25, width = "SJ-dpi", n=25), lty=1, type = "o")
many.med10 <- replicate(1000, one.med(10))
```

```
truehist(many.med10, main = "Estimated Median School Size (n=10)",
xlab = "Mean School Size", ylab = "Frequency", col="#00A396")
lines(density(many.med10, width = "SJ-dpi", n=10), lty=1, type = "o")

###################GRAPH GENERATION#############################
pdf(file="medians.pdf", height=8, width=10)
par(mfrow=c(2,2), bg="gray99")
truehist(many.med100, main = "Estimated Median School Size (n=100)",
xlab = "Median School Size", ylab = "Frequency", col="#C87A8A")
lines(density(many.med100, width = "SJ-dpi", n=100), lty=1, type = "o"
truehist(many.med50, main = "Estimated Median School Size (n=50)",
xlab = "Median School Size", ylab = "Frequency", col="#AC8C4E")
lines(density(many.med50, width = "SJ-dpi", n=50), lty=1, type = "o")
truehist(many.med25, main = "Estimated Median School Size (n=25)",
xlab = "Median School Size", ylab = "Frequency", col="#6B9D59")
lines(density(many.med25, width = "SJ-dpi", n=25), lty=1, type = "o")
truehist(many.med10, main = "Estimated Median School Size (n=10)",
xlab = "Median School Size", ylab = "Frequency", col="#00A396")
lines(density(many.med10, width = "SJ-dpi", n=10), lty=1, type = "o")
dev.off()

############################################################
######                  Problem 2.7C            #############
############################################################

strat_rows100 <- stratsample(calapi$stype, counts=c(E=50, M=25,H=25))
strat_data100 <- calapi[strat_rows100,]
many.meanstrat100<- replicate(1000, one.mean(100))
strat_rows52 <- stratsample(calapi$stype, counts=c(E=26, M=13,H=13))
strat_data52 <- calapi[strat_rows52,]
many.meanstrat52 <- replicate(1000, one.mean(50))
strat_rows24 <- stratsample(calapi$stype, counts=c(E=12, M=6,H=6))
strat_data24 <- calapi[strat_rows24,]
many.meanstrat24 <- replicate(1000, one.mean(24))
```

```
strat_rows12 <- stratsample(calapi$stype, counts=c(E=6, M=3,H=3))
strat_data12 <- calapi[strat_rows12,]
many.meanstrat12 <- replicate(1000, one.mean(12))

##########Graph Making ####################################

pdf(file="stratmean.pdf", height=8, width=10)
par(mfrow=c(2,2), bg="gray99")
truehist(many.meanstrat100, main = "Estimated Stratified Mean School S
xlab = "Mean School Size", ylab = "Frequency", col="#C87A8A")
lines(density(many.meanstrat100, width = "SJ-dpi", n=100), lty=1, type
truehist(many.meanstrat52, main = "Estimated Stratified Mean School Si
xlab = "Mean School Size", ylab = "Frequency", col="#AC8C4E")
lines(density(many.meanstrat52, width = "SJ-dpi", n=52), lty=1, type =
truehist(many.meanstrat24, main = "Estimated Stratified Mean School Si
xlab = "Mean School Size", ylab = "Frequency", col="#6B9D59")
lines(density(many.meanstrat24, width = "SJ-dpi", n=24), lty=1, type =
truehist(many.meanstrat12, main = "Estimated Stratified Mean School Si
xlab = "Meaan School Size", ylab = "Frequency", col="#00A396")
lines(density(many.meanstrat12, width = "SJ-dpi", n=12), lty=1, type =
dev.off()

#############################################################
####                  Problem 3.6                  ##########
#############################################################
library(RSQLite)
library(gmodels)
load(url("http://www.stat.washington.edu/handcock/529/Data/cjis.RData"
fix(cjis2004)
dim(cjis2004)
names(cjis2004)

#############################################################
##                 EXPLORATORY STATISTICS                  ##
```

```
#############################################################

sum(cjis2004$Murder.Total)  # 189 Murders
sum(cjis2004$Burglary.Total)  # 59,985 Burglaries
names(cjis2004)

#############################################################
####                  Problem 3.6 A                  ##########
#############################################################

cl = cluster(cjis2004, "County", 10, method="srswor")
clusterdata = getdata(cjis2004, cl)
dim(clusterdata)
clusterdata$ctsize <- rep(39,61)
set.seed(1)
srs_design <- svydesign(id=~County, fpc=~ctsize, data=clusterdata)
summary(srs_design)
svytotal(~Murder.Total, design=srs_design, na.rm=TRUE)
svytotal(~Burglary.Total, design=srs_design, na.rm=TRUE)

#############################################################
####                  Problem 3.6 B                  ##########
#############################################################

###KING STRAT
king_strat <- subset(cjis2004, County == 'KING')
dim(king_strat) # 38 rows
table(king_strat$County)
table(cjis2004$County)  # Same number of counties

ID_unit<-rep(as.vector(c(300:337)))
ID_unit
Prob <- rep(1, 38)
Prob
```

```
king_strat2 <- cbind(king_strat, ID_unit, Prob)
dim(king_strat2)  # columns ID_unit and Prob added

###NON-KING STRAT
no_king <- subset(cjis2004, County != "KING")
dim(no_king)
table(no_king$County)
five_strat <- cluster(no_king, "County", 5, method="srswor")
five_strat2 <- getdata(no_king, five_strat)
dim(five_strat2)  # 22 / 20
fix(five_strat2)

##Combining King and Non-King Strata
kingfive_strat <- rbind(data.frame(king_strat2), data.frame(five_strat
dim(kingfive_strat) # 60 / 20
fix(kingfive_strat)

##ADDING FPC
kingfive_strat$fpcpop <-rep(39, 60)
dim(kingfive_strat)

##Making Design
cl_design <- svydesign(id=~County, prob=~Prob,
fpc=~fpcpop, data=kingfive_strat)
cl_design

#Estimating Totals
mbtotal <- svytotal(~Murder.Total+Burglary.Total, cl_design, na.rm=T)
mbtotal

###MAKING TABLES IN LATEX
mbtotals <- as.data.frame(svytotal(~Murder.Total+Burglary.Total,
cl_design, na.rm=T))
mbtotals
```

```
latex.table(mbtotals, file="mur.burg3.7b", rowlabel =
"Crime", caption = "Total and Standard Error")

#################################################################
####                    Problem 3.6 c                  ##########
#################################################################

##use king_strat from 3.6b
dim(king_strat)
sample_rows <-sample(1:38,5)
kpolice_srs <- king_strat[sample_rows, ]
dim(kpolice_srs)
fix(kpolice_srs)
ID_unit<-rep(as.vector(c(400:404)))
ID_unit
Prob <- rep(1, 5)
Prob
king_strat3 <- cbind(kpolice_srs, ID_unit, Prob)
dim(king_strat3)  # columns ID_unit and Prob added

#BINDING KING PRECINCTS AND 5 COUNTIES TOGETHER

kpolfive_strat <- rbind(data.frame(king_strat3),
data.frame(five_strat2))
dim(kpolfive_strat) # 27 / 20

#Creating King vs. Other Strata, don't know if I need this
kpolfive_strat$pfstrat <- ifelse(kpolfive_strat$County ==
"KING", c(1),c(2)) #1 = King, 2=Other

##CREATING FPC FOR COUNTY
kpolfive_strat$fpccounty <-rep(39, 27)
##CREATING FPC FOR COUNTY
kpolfive_strat$fpcprec <- rep(246, 27)
```

```
##CREATING PROBABILITY/WEIGHT FOR SECOND STAGE SAMPLING

kingrow <- rep(.13157989,5)
nkingrow <-rep(1,22)

Prob2 <- rbind(c(as.vector(kingrow), as.vector(nkingrow)))
kpolfive_strat2<-cbind(data.frame(kpolfive_strat), as.vector(Prob2))
dim(kpolfive_strat2)
fix(kpolfive_strat2)

######Generating Design

cl2_design <- svydesign(id=~County + Agency, Prob=~Prob + Prob2,
fpc=~ fpccounty + fpcprec, data=kpolfive_strat2)
cl2_design

mbtotal2 <- as.data.frame(svytotal(~Murder.Total+Burglary.Total,
cl2_design, na.rm=T))

###MAKING TABLES IN LATEX
latex.table(mbtotal2, file="murder2", rowlabel = "Crime",
caption = "Murder and Burglary totals and and standard errors.
This clustering approach appears to be significantly less accurate.")

############################################################
####                  Problem 3.6 D                 ##########
############################################################

##Creating Weights for Prob Proportionate to Size
wgt <- cjis2004$Population
wgt[is.na(wgt)] <- 0
sample_rows <- sample(x=1:246, size=10, prob=wgt)
cjis_pps <- cjis2004[sample_rows, ]
cjis_pps$popsize <-rep(39,10)
```

```
cjis_pps$probs <- wgt[sample_rows]
cjis_pps$probs <- cjis_pps$probs/sum(cjis_pps$probs)
list(cjis_pps$Agency)
ppe_design <- svydesign(id=~Agency, fpc=~popsize, prob=~probs,
data=cjis_pps)
mbtotal3 <- as.data.frame(svytotal(~Murder.Total+Burglary.Total,
ppe_design, na.rm=T))
mbtotal3

#################################################################
####                    Problem 3.6 E                  ##########
#################################################################

cl2 = cluster(cjis2004, "County", 20, method="srswor")
clusterdata2 = getdata(cjis2004, cl2)
dim(clusterdata2)
clusterdata2$ctsize <- rep(39,134)
fix(clusterdata2)

set.seed(1)
srs_design <- svydesign(id=~County, fpc=~ctsize, Prob=~Probs,
data=clusterdata2)
summary(srs_design)
mbtotal4 <- as.data.frame(svytotal(~Murder.Total+Burglary.Total,
srs_design, na.rm=T))
mbtotal4
```