```r
data = read.table("/Users/shuhualiang/Documents/Davis/STA 144/
 ProjectData.txt")

data <- data[-1,]

names(data) =
 c("Dis","Add","HoVal","TwelUp","ElevDown","TeleSets","Cable","TVhrs","NewsHr
 s","Sports","ChildPro","Movies")

Area = rep("Rural",length(data[,1]))
Area[data$Dis %in% 51:75] = "Lockhart"
Area[data$Dis %in% 47:50] = "Eavesville"
Area[data$Dis %in% 44] = "Villegas"
Area[data$Dis %in% 45] = "Weldon"
Area[data$Dis %in% 46] = "Routledge"
data$Area = Area

#### Ch2 Prob 5
Lockhart = data[which(data$Area == 'Lockhart'),]

set.seed(2000)
Index = sample(1:length(Lockhart[,1]),200,replace = FALSE)
Sample.Lock = Lockhart[Index,]

N = length(Lockhart$Cable)
n = length(Sample.Lock$Cable)

## a
S.L.Cable.mean = mean(Sample.Lock$Cable)
S.L.Cable.mean

s2.Cable = var(Sample.Lock$Cable)
SE_Cable.mean = sqrt((1-n/N)*s2.Cable/n)
SE_Cable.mean

## b
S.L.TeleSets.mean = mean(Sample.Lock$TeleSets)
S.L.TeleSets.mean

s2.TeleSets = var(Sample.Lock$TeleSets)
SE_TeleSets.mean = sqrt((1-n/N)*s2.TeleSets/n)
SE_TeleSets.mean

## c
Cable.atleast10 = as.numeric(Sample.Lock$Cable >= 10)
Prop.Cable.atleast10 = sum(Cable.atleast10)/length(Sample.Lock$Cable)
Prop.Cable.atleast10

s2.Cable.atleast10 = var(Cable.atleast10)
SE_Prop.Cable.atleast10 = sqrt((1-n/N)*s2.Cable.atleast10/n)
```

```r
SE_Prop.Cable.atleast10


#### Ch2 Prob 6
AssessedVal.mean = mean(as.numeric(as.matrix(Sample.Lock$HoVal)))
AssessedVal.mean

s2.HoVal = var(Sample.Lock$HoVal)
SE_AssessedVal.mean = sqrt((1-n/N)*s2.HoVal/n)
SE_AssessedVal.mean

## 95% CI
AssessedVal.mean - 1.96*SE_AssessedVal.mean
AssessedVal.mean + 1.96*SE_AssessedVal.mean

## Includes the known value $71117


#### Ch3 Prob 10

## Stratify by TwelUp
Lockhart$TwelUp[Lockhart$TwelUp %in% 5:12] = 5
unique(Lockhart$TwelUp)

Strata = split(Lockhart,Lockhart$TwelUp)

Nh = sapply(1:length(Strata),function(i){
    dim(Strata[[i]])[1]
})

nh = ceiling((Nh/N)*200)

S.Sample = lapply(1:length(nh),function(i){
    SampleIndex = sample(1:Nh[[i]],nh[i])
    Data = Strata[[i]][SampleIndex,]
})

Strata1 = do.call(rbind, S.Sample)

##
S.L.Cable.mean.str = mean(Strata1$Cable)
S.L.Cable.mean.str

s2.Cable.STR = array(0,length(Strata))
for(i in 1:length(Strata)){
    s2.Cable.STR[i] = var(S.Sample[[i]]$Cable)
}
s2.Cable.STR

V.h.y.h.cable.str = sum((1-nh/Nh)*Nh^2*s2.Cable.STR/nh)
```

```
SE.y.h.cable.str = sqrt(V.h.y.h.cable.str)/(N^2)
SE.y.h.cable.str


##
S.L.TeleSets.mean = mean(Strata1$TeleSets)
S.L.TeleSets.mean

s2.TeleSets.STR = array(0,length(Strata))
for(i in 1:length(Strata)){
    s2.TeleSets.STR[i] = var(S.Sample[[i]]$TeleSets)
}
s2.TeleSets.STR

V.h.y.h.TeleSets.str = sum((1-nh/Nh)*Nh^2*s2.TeleSets.STR/nh)
SE.y.h.TeleSets.str = sqrt(V.h.y.h.TeleSets.str)/(N^2)
SE.y.h.TeleSets.str


#### Ch3 Prob 11

Sample.Lock$TwelUp[Sample.Lock$TwelUp >= 5] = 5

Str1 = split(Sample.Lock,Sample.Lock$TwelUp)
Str = do.call(rbind,Str1)

##
Again.Cable.mean.str = mean(Str$Cable)
Again.Cable.mean.str

s2.Again.Cable.STR = array(0,length(Str1))
for(i in 1:length(Str1)){
    s2.Again.Cable.STR[i] = var(Str1[[i]]$Cable)
}
s2.Again.Cable.STR

#### Ch3 Prob 12

Co = 20
Cl = 13

nh.opt.ney = ceiling(Nh*sqrt(s2.Again.Cable.STR)*200/
 sum(Nh*sqrt(s2.Again.Cable.STR)))

Opt.Sample = lapply(1:length(nh.opt.ney),function(i){
    SampleIndex = sample(1:Nh[[i]],nh.opt.ney[i])
    Data = Strata[[i]][SampleIndex,]
})

Opt.Sample.list = do.call(rbind,Opt.Sample)
```

```r
Cable.opt.mean = mean(Opt.Sample.list$Cable)
Cable.opt.mean

TeleSets.opt.mean = mean(Opt.Sample.list$TeleSets)
TeleSets.opt.mean


#### Ch3 Prob 14

Var.opt.sample.list = var(Opt.Sample.list$Cable)
Var.opt.sample.list

S.str = sqrt(Var.opt.sample.list)
S.str

e = 1.96*sqrt((1-n/N)/n)*S.str

no = (1.96*sqrt(s2.Cable)/e)^2

n.srs = ceiling(no/(1+no/N))
n.srs

####### Proportional Allocation

Opt.Sample.prop = lapply(1:length(nh),function(i){
    SampleIndex = sample(1:Nh[[i]],nh[i])
    Data = Strata[[i]][SampleIndex,]
})

Opt.Sample.prop.list = do.call(rbind,Opt.Sample.prop)

Var.opt.sample.prop.list = var(Opt.Sample.prop.list$Cable)
Var.opt.sample.prop.list

S.str.prop = sqrt(Var.opt.sample.prop.list)
S.str.prop

e.prop = 1.96*sqrt((1-n/N)/n)*S.str.prop

no.prop = (1.96*sqrt(s2.Cable)/e.prop)^2

n.srs.prop = ceiling(no.prop/(1+no.prop/N))
n.srs.prop

#### Ch4 Prob 16

xU.bar = mean(as.numeric(as.matrix(Lockhart$HoVal)))
Sample.Lock$HoVal = as.numeric(as.matrix(Sample.Lock$HoVal))
```

```r
## a
b.Cable.hat = S.L.Cable.mean/AssessedVal.mean
b.Cable.hat

y_bhr.Cable = b.Cable.hat*xU.bar
y_bhr.Cable

ei.Cable = Sample.Lock$Cable - b.Cable.hat*Sample.Lock$HoVal

Se2.Cable = sum(ei.Cable^2)/(n-1)


vh.y_bhr.Cable = (1-n/N)*(xU.bar/AssessedVal.mean)^2*Se2.Cable/n
SE.Cable.ratio = sqrt(vh.y_bhr.Cable)
SE.Cable.ratio

Sample.Lock$Cable = as.numeric(as.matrix(Sample.Lock$Cable))

plot(Sample.Lock$HoVal, Sample.Lock$Cable, xlab = "House Assessed Value",
 ylab = "Cable Service Cost", main = "House Assessed Value vs. Cable Service
 Cost")

abline(0, b.Cable.hat)

## b
b.TeleSets.hat = S.L.TeleSets.mean/AssessedVal.mean
b.TeleSets.hat

y_bhr.TeleSets = b.TeleSets.hat*xU.bar
y_bhr.TeleSets

ei.TeleSets = Sample.Lock$TeleSets - b.TeleSets.hat*Sample.Lock$HoVal
Se2.TeleSets = sum(ei.TeleSets^2)/(n-1)

vh.y_bhr.TeleSets = (1-n/N)*(xU.bar/AssessedVal.mean)^2*Se2.TeleSets/n
SE.TeleSets.ratio = sqrt(vh.y_bhr.TeleSets)
SE.TeleSets.ratio

Sample.Lock$TeleSets = as.numeric(as.matrix(Sample.Lock$TeleSets))

plot(Sample.Lock$HoVal, Sample.Lock$TeleSets, xlab = "House Assessed Value",
 ylab = "Number of TV Sets", main = "House Assessed Value vs. Number of TV
 Sets")

abline(0, b.TeleSets.hat)

## c
b.Prop.Cable.atleast10.hat = Prop.Cable.atleast10/AssessedVal.mean
b.Prop.Cable.atleast10.hat
```

```
y_bhr.Prop.Cable.atleast10 = b.Prop.Cable.atleast10.hat*xU.bar
y_bhr.Prop.Cable.atleast10

ei.Prop.Cable.atleast10 = Cable.atleast10 -
 b.Prop.Cable.atleast10.hat*Sample.Lock$HoVal
Se2.Prop.Cable.atleast10 = sum(ei.Prop.Cable.atleast10^2)/(n-1)

vh.y_bhr.Prop.Cable.atleast10 = (1-n/N)*(xU.bar/
 AssessedVal.mean)^2*Se2.Prop.Cable.atleast10/n
SE.TeleSets.ratio = sqrt(vh.y_bhr.Prop.Cable.atleast10)
SE.TeleSets.ratio

plot(Sample.Lock$HoVal, Cable.atleast10, xlab = "House Assessed Value", ylab
 = "Proportion", main = "House Assessed Value vs. Proportion of households
 willing to pay at least $10 for Cable TV")

abline(0,b.Prop.Cable.atleast10.hat)


#### Ch4 Prob 17
Lockhart.orig = data[which(data$Area == 'Lockhart'),]

set.seed(2000)
Index.org = sample(1:length(Lockhart[,1]),200,replace = FALSE)
Sample.Lock.org = Lockhart[Index,]

x.bar = mean(Cable.atleast10)
ui = Cable.atleast10*Sample.Lock.org$TwelUp

tx = sum(Cable.atleast10)
tu = sum(ui)

y.bar_d = tu/tx

yi.d = split(Sample.Lock.org$TwelUp,Cable.atleast10)$`1`

s2_yd = sum((yi.d - y.bar_d)^2)/tx
s2_yd

SE.y.bar_d = sqrt((1-n/N)*s2_yd/tx)

CV = SE.y.bar_d/y.bar_d


#### Ch4 Prob 18
total.x = sum(U.Cable.atleast10)

ty.ratio = y.bar*total.x/x.bar
ty.ratio
```

```r
V.tyr_hat = (1-n/N)*(total.x/x.bar)^2*Se2/n
V.tyr_hat

SE.tyr_hat = sqrt(V.tyr_hat)
SE.tyr_hat

CV.total.y = SE.tyr_hat/ty.ratio
CV.total.y


#### Ch5 Prob 20
## Estimate the cost of a sample of 100 addresses from rural districts

set.seed(2003)
Costs = sapply(1:10,function(i){
    SampleIndex = sample(1:length(data[which(as.numeric(as.matrix(data$Dis))
 < 44),1]), 100, replace = FALSE)
    SampleData = data[SampleIndex,]
    TotalCost = 60*length(unique(SampleData$Dis))+16*100
    return(TotalCost)
})

Cost.100 = mean(Costs)

### choose (randomly) 13 districts
N = 43
n = 13
Co.rural = 60
C.rural.ind = 16

Dis.Sampling.Cost = nDis*Co.rural
Dis.Sampling.Cost

Money.left.ind = Cost.100 - Dis.Sampling.Cost
Money.left.ind

m = floor(Money.left.ind/C.rural.ind)
m

### Design
Rural = data[which(as.numeric(as.matrix(data$Dis)) < 44),]
AllMi = aggregate(Rural[,2], by = list(Rural$Dis),length)$x  ## x gives
 frequency

MyDis = sample(1:43,nDis)
Mi = AllMi[MyDis]
Mo = length(Rural[,1])   #### Note: N = Mo

mi = round((Mi/sum(Mi))*m)
```

```r
## Check if withi budget --> OK
Co.rural*nDis + C.rural.ind*sum(mi)

## Create the dataset
SplitDis = split(Rural,as.integer(Rural$Dis))
ClusterSample_1 = lapply(1:nDis,function(i){
    SizeOfDis = nrow(SplitDis[[MyDis[i]]])
    Index = sample(1:SizeOfDis, mi[i])
    Data = SplitDis[[MyDis[[i]]]][Index,]
    return(Data)
})
ClusterSample = do.call(rbind,ClusterSample_1)


#### Ch5 Prob 21

## Unbias Estimate
yi.mean = sapply(1:nDis,function(i){
    mean(ClusterSample_1[[i]]$Cable)
})

ti.hat = Mi*yi.mean
t.hat.unb = N/n*sum(ti.hat)
t.hat.unb

y.bh.unb = t.hat.unb/Mo
y.bh.unb

St2 = sum((ti.hat-t.hat.unb/N)^2)/(n-1)

Si2 = sapply(1:nDis,function(i){
    sum((ClusterSample_1[[i]]$Cable - yi.mean[i])^2)/(mi[i]-1)
})


V.t.hat.unb = N^2*(1-n/N)*St2/n + N*sum((1-mi/Mi)*Mi^2*Si2/mi/n)

SE.y_bh.unb = sqrt(V.t.hat.unb)/Mo
SE.y_bh.unb

## Ratio Esitmate
Y_bh.r = sum(ti.hat)/sum(Mi)
Y_bh.r

Sr2 = sum((ti.hat[i]-Mi[i]*Y_bh.r)^2)/(n-1)

Vhat.y_bh.r = (1/mean(Mi)^2)*(1-n/N)*Sr2/n + sum(Mi^2*(1-mi/Mi)*Si2/mi)/
 (n*N*mean(Mi)^2)
Vhat.y_bh.r
```

```r
SE.y_bh.r = sqrt(Vhat.y_bh.r)
SE.y_bh.r


#### Ch6 Prob 22
SizeWeights = aggregate(Rural$Add,by = list(Rural$Dis),length)$x
ToBeSamp = sample(1:43, 13,replace = TRUE, prob = SizeWeights/
 sum(SizeWeights))

mi.rep = rep(floor(m/nDis),nDis)

## Create the dataset
SplitDis = split(Rural,as.integer(Rural$Dis))
ClusterSample.rep_1 = lapply(1:nDis,function(i){
    SizeOfDis = nrow(SplitDis[[ToBeSamp[i]]])
    Index = sample(1:SizeOfDis, mi.rep[i])
    Data = SplitDis[[ToBeSamp[[i]]]][Index,]
    return(Data)
})
ClusterSample.rep = do.call(rbind,ClusterSample.rep_1)

Psi_i = Mi/Mo
Psi_i

y.bar_ij = sapply(1:nDis,function(i){
    mean(ClusterSample.rep_1[[i]]$Cable)
})

t_hat_i = y.bar_ij*Mi
t_hat_i

y_hat_bar_Psi = mean(t_hat_i/Psi_i)/Mo
y_hat_bar_Psi

Se.y_hat_bar_Psi = sqrt(var(t_hat_i/Psi_i))/Mo
Se.y_hat_bar_Psi
```