STAT410 Project

Jiajun Zhang, LingXiang Zou, Tan Tian

March 25, 2019

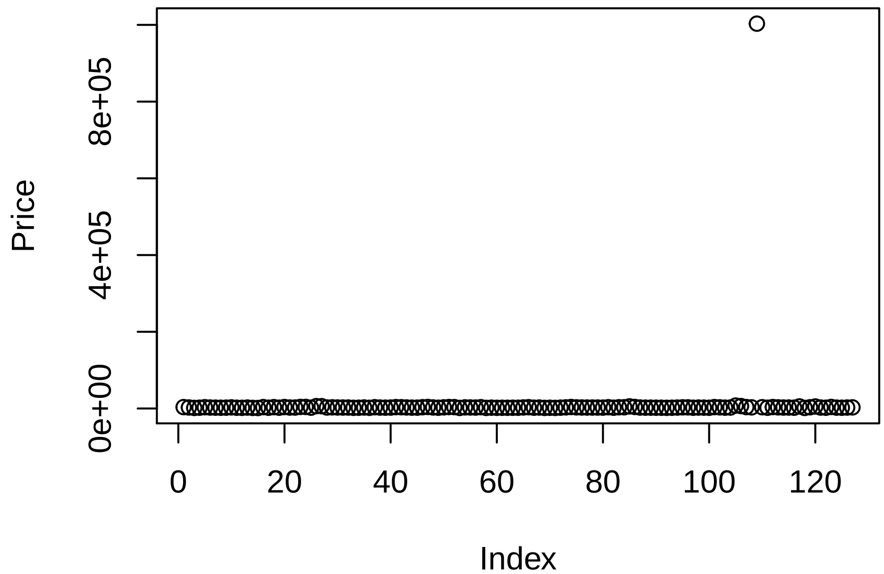
```
#data cleaning, Tokyo hostels as population
suppressPackageStartupMessages(library(tidyverse))
hostel=read.csv("japanHostel.csv")

Tokyo=filter(hostel,City=="Tokyo")
head(Tokyo)
```

```
##   X                 hostel.name  City price.from                Distance
## 1 3      &And Hostel Akihabara Tokyo       3600  7.8km from city centre
## 2 4           &And Hostel Ueno Tokyo       2600  8.7km from city centre
## 3 5 &And Hostel-Asakusa North- Tokyo       1500 10.5km from city centre
## 4 6     1night1980hostel Tokyo Tokyo       2100  9.4km from city centre
## 5 7         328 Hostel & Lounge Tokyo       3300 16.5km from city centre
## 6 9   3Q House - Asakusa Smile Tokyo       2500 10.2km from city centre
##   summary.score rating.band atmosphere cleanliness facilities location.y
## 1           8.7    Fabulous        8.0         7.0        9.0        8.0
## 2           7.4   Very Good        8.0         7.5        7.5        7.5
## 3           9.4      Superb        9.5         9.5        9.0        9.0
## 4           7.0   Very Good        5.5         8.0        6.0        6.0
## 5           9.3      Superb        8.7         9.7        9.3        9.1
## 6            NA        <NA>         NA          NA         NA         NA
##   security staff valueformoney      lon      lat
## 1     10.0  10.0           9.0 139.7775 35.69745
## 2      7.0   8.0           6.5 139.7837 35.71272
## 3      9.5  10.0           9.5 139.7984 35.72790
## 4      8.5   8.5           6.5 139.7869 35.72438
## 5      9.3   9.7           8.9 139.7455 35.54804
## 6       NA    NA            NA       NA       NA
```

```
#Check NA
c( table(is.na(Tokyo$price.from)), table(is.na(Tokyo$Distance)) )
```

```
## FALSE FALSE
##   127   127
```

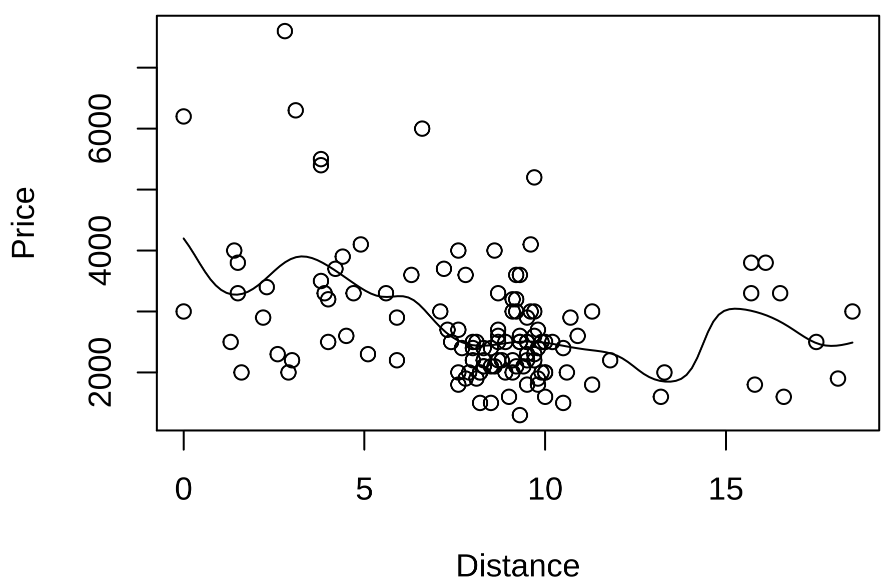
```
Tokyo$Distance=as.numeric(gsub("km.*","",Tokyo$Distance))
Tokyo=Tokyo%>%rename(Price=price.from)%>%select(Price, Distance)

#From the plot we see there is a obvious outlier
plot(Tokyo$Price, ylab="Price")
```




```
#Remove obs 109 where the price is considered as an outlier
Tokyo=Tokyo[-109,]

#Check the extreme of distance
c( max(Tokyo$Distance), min(Tokyo$Distance) )
```

```
## [1] 18.5  0.0
```

```
#There is sort of relationship b/w dist and price, but not quite, fit lm model
plot(Tokyo$Distance, Tokyo$Price, xlab="Distance", ylab="Price")
lines(ksmooth(Tokyo$Distance, Tokyo$Price, kernel="normal", bandwidth=2))
```

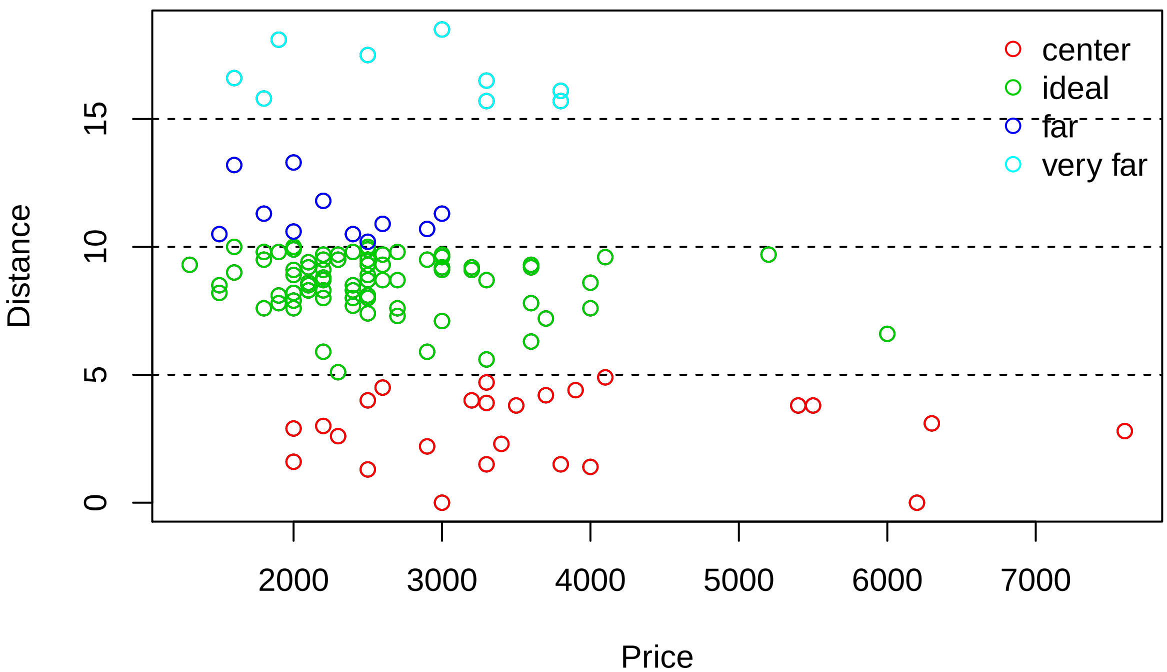



```
dis <- Tokyo$Distance
pric <- Tokyo$Price
lm1 <- lm(dis~pric)
summary(lm1)
```

```
##
```

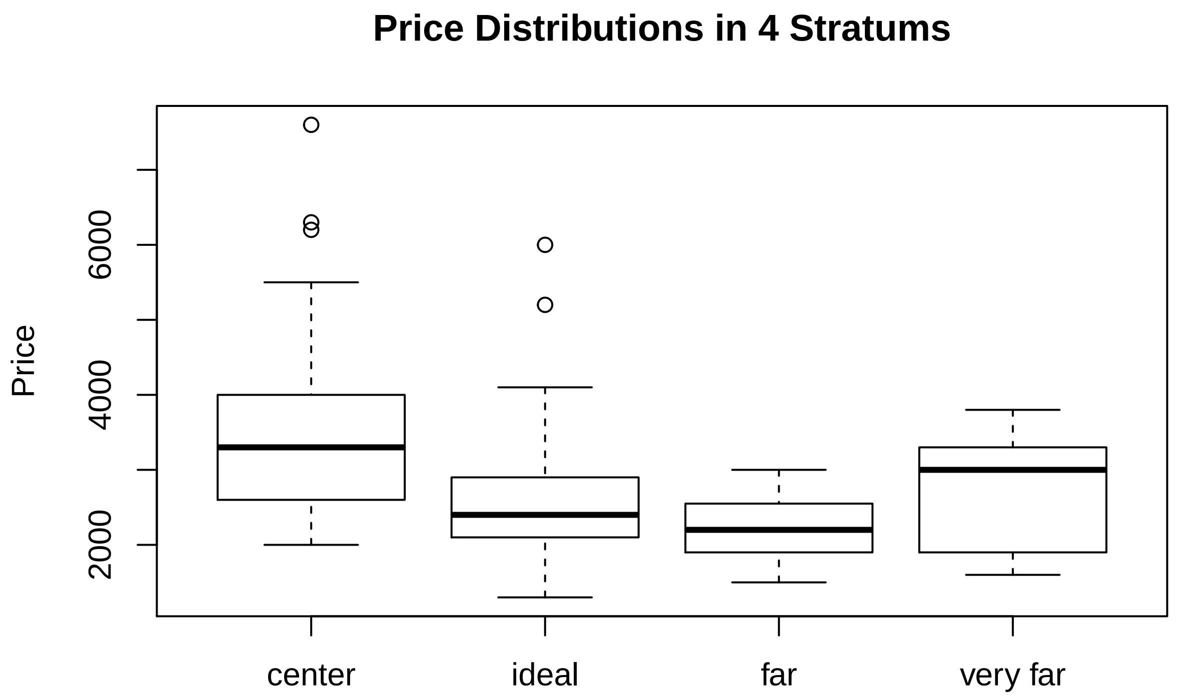
```
## Call:
## lm(formula = dis ~ pric)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0170 -1.2773  0.0771  1.3136 10.4830
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.6851653  0.8644199  13.518  < 2e-16 ***
## pric        -0.0012227  0.0002922  -4.185 5.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.408 on 124 degrees of freedom
## Multiple R-squared:  0.1237, Adjusted R-squared:  0.1167
## F-statistic: 17.51 on 1 and 124 DF,  p-value: 5.365e-05
```

```r
#Reverse the axis to visualize stratums
plot(y=Tokyo$Distance, x=Tokyo$Price, xlab="Price", ylab="Distance")
abline(h=c(5,10,15), lty=2)
points(x=Tokyo$Price[Tokyo[,"Distance"]<=5], y=Tokyo$Distance[Tokyo[,"Distance"]<=5], col=2)
points(x=Tokyo$Price[Tokyo[,"Distance"]>5&Tokyo[,"Distance"]<=10],
       y=Tokyo$Distance[Tokyo[,"Distance"]>5&Tokyo[,"Distance"]<=10], col=3)
points(x=Tokyo$Price[Tokyo[,"Distance"]>10&Tokyo[,"Distance"]<=15],
       y=Tokyo$Distance[Tokyo[,"Distance"]>10&Tokyo[,"Distance"]<=15], col=4)
points(x=Tokyo$Price[Tokyo[,"Distance"]>15], y=Tokyo$Distance[Tokyo[,"Distance"]>15], col=5)
legend("topright", col=c(2,3,4,5), pch=c(1,1,1,1),
       c("center", "ideal", "far", "very far"), bty="n")
```




```r
y1=Tokyo$Price[Tokyo[,"Distance"]<=5]; y2=Tokyo$Price[Tokyo[,"Distance"]>5&Tokyo[,"Distance"]<=10]
y3=Tokyo$Price[Tokyo[,"Distance"]>10&Tokyo[,"Distance"]<=15]; y4=Tokyo$Price[Tokyo[,"Distance"]>15]
```

```
#Check the distributions of all 4 stratums
boxplot(y1,y2,y3,y4, names=c("center", "ideal", "far", "very far"), ylab="Price",
        main="Price Distributions in 4 Stratums")
```




```
Tokyo=Tokyo%>%mutate(strata = cut(Distance,breaks=c(-0.1,5,10,15,20)))
Tokyo=mutate(Tokyo, strata = recode_factor(strata,
                                             "(-0.1,5]" = "center",
                                             "(5,10]" = "ideal",
                                             "(10,15]" = "far",
                                             "(15,20]" = "very far"))
head(Tokyo)
```

```
##   Price Distance   strata
## 1  3600      7.8    ideal
## 2  2600      8.7    ideal
## 3  1500     10.5      far
## 4  2100      9.4    ideal
## 5  3300     16.5 very far
## 6  2500     10.2      far
```

```
mu=mean(Tokyo$Price)
```

SRS

```
y=Tokyo$Price
N=length(y)
n=50
ybar=NULL;sv=NULL
```

```r
for(i in 1:10000){
  s=sample(1:N,n,r=F) #without replacement
  ybar[i]=mean(y[s])
  sv[i]=var(y[s])

}


low=ybar-qt(0.975,d=n-1)*sqrt((1-n/N)*sv/n)
up=ybar+qt(0.975,d=n-1)*sqrt((1-n/N)*sv/n)

# compute the coverage probability
cover_prob=sum( (low<=mu)*(up>=mu) )/10000
cover_prob
```

```
## [1] 0.9399
```

```r
ybar_rT=NA
for(i in 1:10000){
  s_r=sample(1:N,n,r=T) #with replacement
  ybar_rT[i]=mean(y[s_r])
}
mean((ybar_rT-mu)^2)
```

```
## [1] 21909.76
```

Unequal Probability Random Sampling with Replacement

```r
#Assign prob using the 1/x form because we expect closer dist has higher prob
x=Tokyo$Distance

for(i in seq_along(x)){
  if(x[i]==0){
    #Set this as second min, because 1/0=Inf
    x[i]=min(x[x!=min(x)])
  }
}

p=(1/x)/sum(1/x)

ybar_unprobb=NA; mu_HH=NA; mu_HT=NA; mu_GUPE=NA
pi=1-(1-p)^n
for(i in 1:10000){
  ss=sample(1:N, n, r=T, prob=p)
  mu_HH[i]=mean(y[ss]/p[ss])/N
  ssu=unique(ss)
  mu_HT[i]=sum(y[ssu]/pi[ssu])/N
  mu_GUPE[i]=sum(y[ss]/pi[ss])/sum(1/pi[ss])
}

#Mean
```

```
c( True_mean=mu, HH_estimate=mean(mu_HH), HT_estimate=mean(mu_HT),
   GUPE_estimate=mean(mu_GUPE))
```

```
##     True_mean   HH_estimate   HT_estimate GUPE_estimate
##      2769.841      2771.171      2770.912      2837.382
```

```
#MSE
c( SRS=mean((ybar-mu)^2), HH=mean((mu_HH-mu)^2), HT=mean((mu_HT-mu)^2),
   GUPE=mean((mu_GUPE-mu)^2))
```

```
##      SRS       HH       HT     GUPE
## 12802.16 61328.97 81861.51 27237.49
```

Stratified Random Sampling, Proportional Allocation

```
c( Popn_mean=mean(Tokyo$Price), Popn_var=var(Tokyo$Price) )
```

```
##   Popn_mean    Popn_var
##    2769.841 1088363.175
```

```
(strata_table=table(Tokyo$strata))
```

```
##
##   center    ideal      far very far
##       25       81       11        9
```

```
N1=as.numeric(strata_table[1]); N2=as.numeric(strata_table[2])
N3=as.numeric(strata_table[3]); N4=as.numeric(strata_table[4])
n=50

#Proportional
(prop_table=round((table(Tokyo$strata)/nrow(Tokyo)*n), 0))
```

```
##
##   center    ideal      far very far
##       10       32        4        4
```

```
n_p1=as.numeric(prop_table[1]); n_p2=as.numeric(prop_table[2])
n_p3=as.numeric(prop_table[3]); n_p4=as.numeric(prop_table[4])


#Stratified Random Sampling(Prop Allocation)
ybar_strat=NA;var_strat=NA

for(i in 1:10000){
  s1=sample(1:N1, n_p1)
  s2=sample(1:N2, n_p2)
  s3=sample(1:N3, n_p3)
  s4=sample(1:N4, n_p4)
  # sum(N_h*ybar_h)/N
  ybar_strat[i]=( (mean(y1[s1])*N1)+(mean(y2[s2])*N2)+(mean(y3[s3])*N3)+(mean(y4[s4])*N4) )/N
  var_strat[i]=( ((N1/N)^2)*( ((N1-n_p1)/N1)*(var(y1[s1]))/n_p1)+
                 ((N2/N)^2)*( ((N2-n_p2)/N2)*(var(y2[s2]))/n_p2)+
                 ((N3/N)^2)*( ((N3-n_p3)/N3)*(var(y3[s3]))/n_p3)+
```

```
                  ((N4/N)^2)*( ((N4-n_p4)/N4)*(var(y4[s4]))/n_p4))
}

low_strat=ybar_strat-qt(0.975,d=n-1)*sqrt(var_strat)
up_strat=ybar_strat+qt(0.975,d=n-1)*sqrt(var_strat)

# compute the coverage probability
cover_prob_strat=sum( (low_strat<=mu)*(up_strat>=mu) )/10000
cover_prob_strat
```

```
## [1] 0.9375
```

```
#Mean comparison
c( True_Popn=mu, SRS=mean(ybar), Stratified_PA=mean(ybar_strat) )
```

```
##     True_Popn           SRS Stratified_PA
##      2769.841      2771.149      2768.189
```

```
#MSE comparison
c( SRS=mean((ybar-mu)^2), Stratified_PA=mean((ybar_strat-mu)^2) )
```

```
##           SRS Stratified_PA
##      12802.16      10619.78
```

Stratified Random Sampling, Optimal Allocation

```
#Optimal
#var of within each stratum
sigma_sq=tapply(Tokyo$Price, Tokyo$strata, var)
sigma_sq_1=as.numeric(sigma_sq[1]); sigma_sq_2=as.numeric(sigma_sq[2])
sigma_sq_3=as.numeric(sigma_sq[3]); sigma_sq_4=as.numeric(sigma_sq[4])
sigma1=sqrt(sigma_sq_1); sigma2=sqrt(sigma_sq_2)
sigma3=sqrt(sigma_sq_3); sigma4=sqrt(sigma_sq_4)
#standard deviation within each stratum
#c(sigma1,sigma2,sigma3,sigma4)

n_o1=round( (n*N1*sigma1)/sum((N1*sigma1)+(N2*sigma2)+(N3*sigma3)+(N4*sigma4)), 0)
n_o2=round( (n*N2*sigma2)/sum((N1*sigma1)+(N2*sigma2)+(N3*sigma3)+(N4*sigma4)), 0)
n_o3=round( (n*N3*sigma3)/sum((N1*sigma1)+(N2*sigma2)+(N3*sigma3)+(N4*sigma4)), 0)
n_o4=round( (n*N4*sigma4)/sum((N1*sigma1)+(N2*sigma2)+(N3*sigma3)+(N4*sigma4)), 0)

ybar_strat_Opti=NA;var_strat_opti=NA
#Stratified Random Sampling(Optimal Allocation)
for(i in 1:10000){
    ss1=sample(1:N1, n_o1)
    ss2=sample(1:N2, n_o2)
    ss3=sample(1:N3, n_o3)
    ss4=sample(1:N4, n_o4)
    ybar_strat_Opti[i]=
      ( (mean(y1[ss1])*N1)+(mean(y2[ss2])*N2)+(mean(y3[ss3])*N3)+(mean(y4[ss4])*N4) )/N

    var_strat_opti[i]=( ((N1/N)^2)*( ((N1-n_o1)/N1)*(var(y1[ss1]))/n_o1)+
```

```r
                   ((N2/N)^2)*( ((N2-n_o2)/N2)*(var(y2[ss2]))/n_o2)+
                   ((N3/N)^2)*( ((N3-n_o3)/N3)*(var(y3[ss3]))/n_o3)+
                   ((N4/N)^2)*( ((N4-n_o4)/N4)*(var(y4[ss4]))/n_o4))
}

low_strat_opti=ybar_strat_Opti-qt(0.975,d=n-1)*sqrt(var_strat_opti)
up_strat_opti=ybar_strat_Opti+qt(0.975,d=n-1)*sqrt(var_strat_opti)

# compute the coverage probability
cover_opti_strat=sum( (low_strat_opti<=mu)*(up_strat_opti>=mu) )/10000
cover_opti_strat
```

```
## [1] 0.9445
```

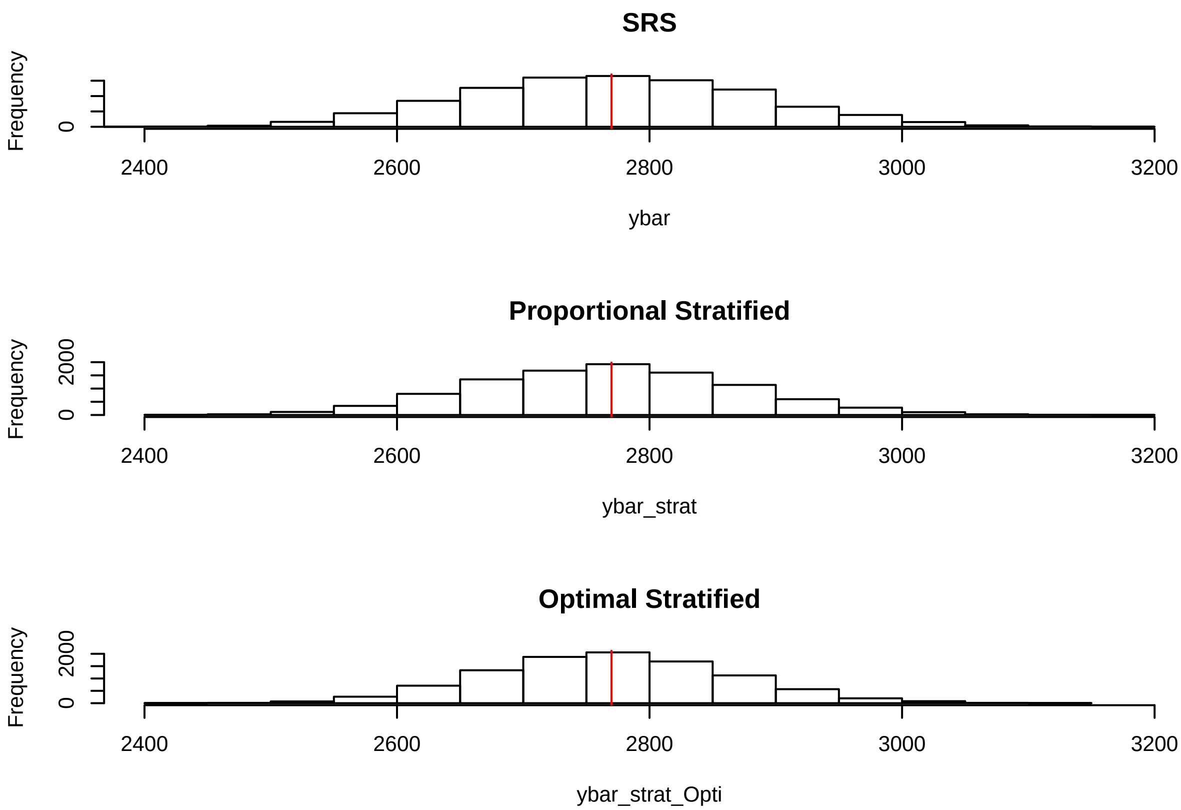
```r
#Mean comparison
c( True_Popn=mu, SRS=mean(ybar), Stratified_PA=mean(ybar_strat),
   Stratified_Opti = mean(ybar_strat_Opti))
```

```
##       True_Popn             SRS   Stratified_PA Stratified_Opti
##        2769.841        2771.149        2768.189        2768.778
```

```r
#MSE comparison
c( SRS=mean((ybar-mu)^2), Stratified_Prop=mean((ybar_strat-mu)^2),
   Stratified_Opti=mean((ybar_strat_Opti-mu)^2) )
```

```
##             SRS Stratified_Prop Stratified_Opti
##        12802.16        10619.78         9042.88
```

```r
par(mfrow=c(3,1))
hist(ybar, xlim=c(2400,3200), main="SRS"); abline(v=mu, col=2)
hist(ybar_strat, xlim=c(2400,3200),
     main="Proportional Stratified"); abline(v=mu, col=2)
hist(ybar_strat_Opti, xlim=c(2400,3200),
     main="Optimal Stratified"); abline(v=mu, col=2)
```



Stratified Sampling With Unequal Probability

```
# Within each stratum, the distances have unequal prob to be selected
x1=Tokyo$Distance[Tokyo[,"Distance"]<=5]
x2=Tokyo$Distance[Tokyo[,"Distance"]>5 & Tokyo[,"Distance"]<=10]
x3=Tokyo$Distance[Tokyo[,"Distance"]>10 & Tokyo[,"Distance"]<=15]
x4=Tokyo$Distance[Tokyo[,"Distance"]>15]

for(i in seq_along(x1)){
  if(x1[i]==0){
    #Set this as second min, because 1/0=Inf
    x1[i]=min(x1[x1!=min(x1)])
  }
}

x1_prob=(1/x1)/sum(1/x1); x2_prob=(1/x2)/sum(1/x2)
x3_prob=(1/x3)/sum(1/x3); x4_prob=(1/x4)/sum(1/x4)
```

```
#HH, HT estimator
pi_1=1-((1-x1_prob)^n_o1); pi_2=1-((1-x2_prob)^n_o2);
pi_3=1-((1-x3_prob)^n_o3); pi_4=1-((1-x4_prob)^n_o4)

mu_strat_unprob_HH=NA; mu_strat_unprob_HT=NA
mu_strat_unprob_GUPE=NA
for(i in 1:10000){
  ss_1=sample(1:N1, n_o1, r=T, prob=x1_prob)
  ss_2=sample(1:N2, n_o2, r=T, prob=x2_prob)
```

```
  ss_3=sample(1:N3, n_o3, r=T, prob=x3_prob)
  ss_4=sample(1:N4, n_o4, r=T, prob=x4_prob)
  #tau_hat/N where tau_hat=sum(tau_hat_h)
  mu_strat_unprob_HH[i]=( mean(y1[ss_1]/x1_prob[ss_1])+mean(y2[ss_2]/x2_prob[ss_2])+
    mean(y3[ss_3]/x3_prob[ss_3])+mean(y4[ss_4]/x4_prob[ss_4]) )/N

  su1=unique(ss_1); su2=unique(ss_2); su3=unique(ss_3); su4=unique(ss_4)
  mu_strat_unprob_HT[i]=( sum(y1[su1]/pi_1[su1])+sum(y2[su2]/pi_2[su2])+
    sum(y3[su3]/pi_3[su3])+sum(y4[su4]/pi_4[su4]) )/N

  #tau_hat_h = mu_hat_g * Nh
  mu_strat_unprob_GUPE[i]=( (sum(y1[ss_1]/pi_1[ss_1])/sum(1/pi_1[ss_1])*N1)+
    (sum(y2[ss_2]/pi_2[ss_2])/sum(1/pi_2[ss_2])*N2)+
    (sum(y3[ss_3]/pi_3[ss_3])/sum(1/pi_3[ss_3])*N3)+
    (sum(y4[ss_4]/pi_4[ss_4])/sum(1/pi_4[ss_4])*N4) )/N

}
```

Regression Estimation

```
n=50
reg_mu=NA;reg_var=NA
mu_x=mean(Tokyo$Distance)

for(i in 1:10000){
  s=sample(1:N,n,r=F) #without replacement
  xi=Tokyo$Distance[s]
  yi=Tokyo$Price[s]
  x_bar=mean(xi)
  y_bar=mean(yi)
  b=sum((xi-x_bar)*(yi-y_bar))/sum((xi-x_bar)^2)
  a=y_bar-(b*x_bar)
  reg_mu[i]=a+(b*mu_x)
  reg_var[i]=( (N-n)/(N*n*(n-2)) )*sum((yi-a-b*xi)^2)
}

low_reg=reg_mu-qt(0.975,d=n-1)*sqrt(reg_var)
up_reg=reg_mu+qt(0.975,d=n-1)*sqrt(reg_var)

# compute the coverage probability
cover_prob_reg=sum( (low_reg<=mu)*(up_reg>=mu) )/10000
cover_prob_reg
```

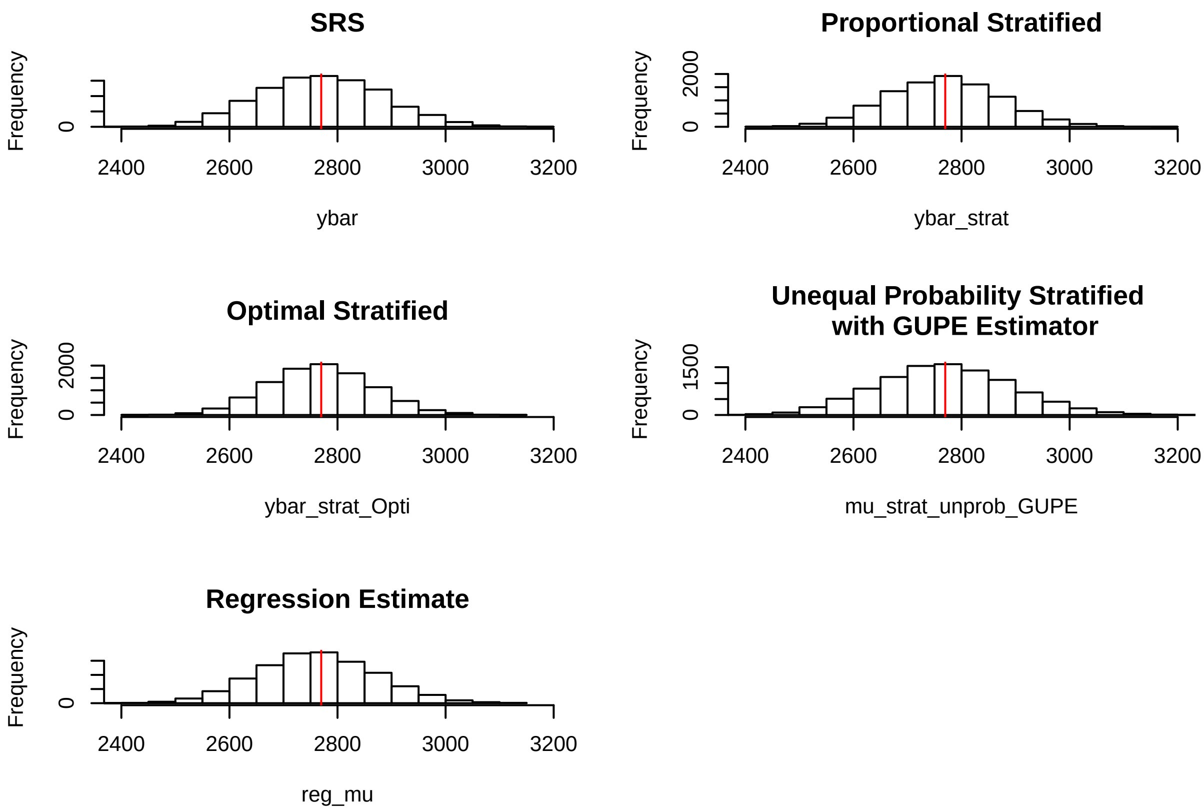
```
## [1] 0.9308
```

```
#Mean comparison
Mean=c(SRS=mean(ybar), Stratified_PA=mean(ybar_strat),
   Stratified_Opti = mean(ybar_strat_Opti), reg_est = mean(reg_mu))

#MSE Comparison
MSE=c( SRS=mean((ybar-mu)^2), Stratified_Prop=mean((ybar_strat-mu)^2),
```

```r
   Stratified_Opti=mean((ybar_strat_Opti-mu)^2),
   reg_est=mean((reg_mu-mu)^2) )

#Bias
Bias=c( SRS=mean(ybar)-mu, Stratified_Prop=mean(ybar_strat)-mu,
   Stratified_Opti=mean(ybar_strat_Opti)-mu,
   reg_est=mean(reg_mu)-mu )

#Coverage of CI comparison
CvgCI=c(SRS=cover_prob, Stratified_Prop=cover_prob_strat,
   Stratified_Opti=cover_opti_strat,
   reg_est=cover_prob_reg  )

#Compare All!
comparison=data.frame(Mean,MSE,Bias,CvgCI,stringsAsFactors=FALSE)
comparison=rbind(comparison,Pop_mean=c(mu,NA,NA,NA))
comparison
```

```
##                     Mean      MSE      Bias  CvgCI
## SRS             2771.149 12802.16  1.308130 0.9399
## Stratified_PA   2768.189 10619.78 -1.651939 0.9375
## Stratified_Opti 2768.778  9042.88 -1.063759 0.9445
## reg_est         2762.293 11861.65 -7.548040 0.9308
## Pop_mean        2769.841       NA        NA     NA
```

```r
#Histograms Visualization
par(mfrow=c(3,2))
hist(ybar, xlim=c(2400,3200), main="SRS"); abline(v=mu, col=2)
hist(ybar_strat, xlim=c(2400,3200),
     main="Proportional Stratified"); abline(v=mu, col=2)
hist(ybar_strat_Opti, xlim=c(2400,3200),
     main="Optimal Stratified"); abline(v=mu, col=2)
hist(mu_strat_unprob_GUPE, xlim=c(2400,3200),
     main="Unequal Probability Stratified \n with GUPE Estimator"); abline(v=mu, col=2)
hist(reg_mu, xlim=c(2400,3200),
     main="Regression Estimate"); abline(v=mu, col=2)
```



As the result, all these designs return good estimates of the average price of the hostels,so we cannot conclude which design performs the best based on the mean estimation.

However, based on other measures, stratified random sampling with optimum allocation returns a smallest mean square error, relatively low bias, and highest probability confidence interval coverage among the four estimations.

Therefore, we can conclude that stratified random sampling with optimum allocation is the most reliable effective design to estimate the population mean.