

STAT 344 Project

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.2      v purrr  1.0.1
## v tibble  3.2.1      v dplyr  1.1.1
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.4      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
dat <- read_csv('crimedata_csv_AllNeighbourhoods_2022.csv')
summary(dat)
```

```
##      TYPE              YEAR      MONTH      DAY
## Length:34279      Min.   :2022      Min.   : 1.000      Min.   : 1.00
## Class :character  1st Qu.:2022      1st Qu.: 4.000      1st Qu.: 8.00
## Mode  :character  Median :2022      Median : 7.000      Median :15.00
##                               Mean  :2022      Mean   : 6.476      Mean   :15.28
##                               3rd Qu.:2022      3rd Qu.: 9.000      3rd Qu.:22.00
##                               Max.   :2022      Max.   :12.000      Max.   :31.00
##
##      HOUR      MINUTE      HUNDRED_BLOCK      NEIGHBOURHOOD
## Min.   : 0.00      Min.   : 0.00      Length:34279      Length:34279
## 1st Qu.: 4.00      1st Qu.: 0.00      Class :character      Class :character
## Median :13.00      Median :10.00      Mode  :character      Mode  :character
## Mean   :11.49      Mean   :17.23
## 3rd Qu.:18.00      3rd Qu.:30.00
## Max.   :23.00      Max.   :59.00
##
##      X      Y
## Min.   :      0      Min.   :      0
## 1st Qu.:490358      1st Qu.:5453670
## Median :491641      Median :5457123
## Mean   :436022      Mean   :4831996
## 3rd Qu.:493147      3rd Qu.:5458733
## Max.   :498296      Max.   :5462300
## NA's   :1          NA's   :1
```

```
#Convert categorical attributes into factor
```

```
dat[c('TYPE', 'HUNDRED_BLOCK', 'NEIGHBOURHOOD')] <- lapply(dat[c('TYPE', 'HUNDRED_BLOCK', 'NEIGHBOURHOOD')],
  function(x) {
    dat <- dat[dat$NEIGHBOURHOOD!="",]
    ( unique(dat$TYPE) )
  })
```

```
## [1] Break and Enter Commercial
## [2] Break and Enter Residential/Other
## [3] Homicide
## [4] Mischief
```

```
## [5] Offence Against a Person
## [6] Other Theft
## [7] Theft from Vehicle
## [8] Theft of Bicycle
## [9] Theft of Vehicle
## [10] Vehicle Collision or Pedestrian Struck (with Fatality)
## [11] Vehicle Collision or Pedestrian Struck (with Injury)
## 11 Levels: Break and Enter Commercial ... Vehicle Collision or Pedestrian Struck (with Injury)
( unique(dat$NEIGHBOURHOOD))
```

```
## [1] West End          Shaughnessy
## [3] Central Business District Grandview-Woodland
## [5] Mount Pleasant    Sunset
## [7] Kensington-Cedar Cottage Strathcona
## [9] Fairview           Oakridge
## [11] Marpole            Kitsilano
## [13] West Point Grey    Victoria-Fraserview
## [15] Hastings-Sunrise   Kerrisdale
## [17] Riley Park         Arbutus Ridge
## [19] Renfrew-Collingwood Killarney
## [21] South Cambie        Dunbar-Southlands
## [23] Stanley Park        Musqueam
## 25 Levels: Arbutus Ridge Central Business District ... West Point Grey
```

```
H <- length(unique(dat$NEIGHBOURHOOD))
summary(dat$TYPE)
```

```
##              Break and Enter Commercial
##                               1984
##      Break and Enter Residential/Other
##                               1266
##                               Homicide
##                               11
##                               Mischief
##                               5613
##      Offence Against a Person
##                               3911
##              Other Theft
##                               10749
##      Theft from Vehicle
##                               7273
##              Theft of Bicycle
##                               1528
##      Theft of Vehicle
##                               910
## Vehicle Collision or Pedestrian Struck (with Fatality)
##                               19
##      Vehicle Collision or Pedestrian Struck (with Injury)
##                               1010
```

Population and taking sample

```
N <- nrow(dat) #Population size
n <- 1000 # Sample size
set.seed(344)
```

```

library(sampling)
#Stratified sample by neighborhood
dat <- dat[order(dat$NEIGHBOURHOOD),]
dat <- dat[-(1:5),] #Drop first 5 rows where the neighborhood name is blank space
#freq <- table(dat$NEIGHBOURHOOD)/nrow(dat)
#freq <- as.vector(freq[-1])
#n.h <- round(freq*n) #Each stratum sample size
# strt <- strata(dat, stratanames = 'NEIGHBOURHOOD', size=n.h, method = 'srswr')

#Grouping of neighborhoods into larger areas
D1 <- c('West End', 'Yaletown', 'Coal Harbour', 'Central Business District', 'Stanley Park')
D2 <- c('Strathcona', 'Grandview-Woodland', 'Hastings-Sunrise', 'Downtown Eastside')
D3 <- c('Sunset', 'Renfrew-Collingwood', 'Mount Pleasant', 'Killarney', 'Victoria-Fraserview', 'Kensington')
D4 <- c('West Point Grey', 'Kitsilano', 'Fairview', 'Dunbar-Southlands', 'Arbutus Ridge', 'Shaughnessy')
dat$DISTRICT[dat$NEIGHBOURHOOD %in% D1] = 'D1'
dat$DISTRICT[dat$NEIGHBOURHOOD %in% D2] = 'D2'
dat$DISTRICT[dat$NEIGHBOURHOOD %in% D3] = 'D3'
dat$DISTRICT[dat$NEIGHBOURHOOD %in% D4] = 'D4'
freq <- table(dat$DISTRICT)/nrow(dat)
freq <- as.vector(freq)
n.h <- round(freq*n)
strt <- strata(dat, stratanames = 'DISTRICT', size=n.h, method = 'srswr')
sample.strt <- dat[strt$ID_unit,]

#Taking an SRS of size n=1000 from the population
set.seed(344)
SRS.index <- sample.int(N, n, replace = FALSE)
SRS <- dat[SRS.index,]

```

Now, we will estimate the total number of crimes that happen during the summer months (July-August) using two samples above. We first use the SRS estimate and report both the estimated value as well as the standard error.

```

SRS$u <- ifelse(SRS$MONTH %in% c(7,8),1,0) #Create dummy for if the month is in summer
tot.hat.SRS <- N*mean(SRS$u)
#Standard error of the estimator, including FPC
se.SRS <- N*sqrt(var(SRS$u)/n)
( summer.SRS.results <- c(tot.hat.SRS, se.SRS) )

```

```
## [1] 5758.0320 405.4136
```

```
#95% CI
```

```
(ci.summer.SRS <- c(tot.hat.SRS-1.96*se.SRS,tot.hat.SRS+1.96*se.SRS))
```

```
## [1] 4963.421 6552.643
```

Next, we will find the stratification estimator:

```

sample.strt$summer <- ifelse(sample.strt$MONTH %in% c(7,8),1,0) #Create dummy for if the month is in summer
N.h <- dat %>% group_by(DISTRICT) %>% count() #population size for the strata

mu.h.str <- sample.strt %>% #number of crimes in summer for each strata
  group_by(NEIGHBOURHOOD) %>%
  summarise(mu.h = mean(summer))
mu.hat <- mean(mu.h.str$mu.h*N.h$n)
tot.hat.str <- H*mu.hat #Estimated value
#Standard error

```

```
se.str <- H*sqrt(var(mu.h.str$mu.h*N.h$n)/n)
( summer.str.results <- c(tot.hat.str,se.str) )
```

```
## [1] 38959.5668 922.2049
```

```
#95% CI
```

```
( ci.summer.str <- c(tot.hat.str-1.96*se.str,tot.hat.str+1.96*se.str))
```

```
## [1] 37152.05 40767.09
```

We will now move on to the second parameter of interest, which is the proportion of Theft of Bicycle out of all crimes in 2022. We will use the same SRS and stratified sample as given above.

```
#Estimator from SRS
```

```
bike_theft <- SRS[SRS$TYPE=='Theft of Bicycle',]
```

```
p.hat.SRS_bike <- nrow(bike_theft)/n
```

```
#Standard error of the estimator, including FPC
```

```
se.SRS_bike <- sqrt((1-n/N)*p.hat.SRS_bike*(1-p.hat.SRS_bike)/n)
```

```
( bike_theft.SRS.results <- c(p.hat.SRS_bike, se.SRS_bike) )
```

```
## [1] 0.037000000 0.005881446
```

```
#95% CI
```

```
( ci.bike.SRS <- c(p.hat.SRS_bike-1.96*se.SRS_bike,p.hat.SRS_bike+1.96*se.SRS_bike))
```

```
## [1] 0.02547237 0.04852763
```

```
#Estimator from stratified sample
```

```
sample.strt$bike_theft <- ifelse(sample.strt$TYPE=='Theft of Bicycle',1,0) #Create dummy for if the crime is Theft of Bicycle
```

```
p.hat.h_bike <- sample.strt %>% #proportion for each strata
```

```
group_by(NEIGHBOURHOOD) %>%
```

```
summarise(p.hat.h = mean(bike_theft))
```

```
p.hat.str_bike <- sum(N.h$n/N*p.hat.h_bike$p.hat.h) #Estimated value
```

```
#Standard error
```

```
se.h_bike <- sqrt((1 - n.h / N.h$n) * p.hat.h_bike$p.hat.h*(1-p.hat.h_bike$p.hat.h) / n.h)
```

```
se.str_bike <- sqrt(sum((N.h$n / N)^2 * se.h_bike^2))
```

```
( bike_theft.str.results <- c(p.hat.str_bike,se.str_bike) )
```

```
## [1] 0.53081723 0.01561585
```

```
#95% CI
```

```
( ci.bike.str <- c(p.hat.str_bike-1.96*se.str_bike, p.hat.str_bike+1.96*se.str_bike) )
```

```
## [1] 0.5002102 0.5614243
```