Problem 1

HW2

Hanyuan Chi(chixx105), Zhi Shen(shenx704)

April 21, 2017

```r
suppressPackageStartupMessages({
  library(purrr)
  library(tidyr)
  library(ggplot2)
  library(dplyr)
})

set.seed(123456) # PLEASE DO NOT CHANGE THE SEED
```

German Tank Problem

You are working for Cool Gadget Inc., a company creating cool electronic gadgets (as the name implies). And, unfortunately, you have a competitor. Moreover, this competitor has recently developed a gadget, ordered and produced a large batch of them and released this batch into the market.

Your company is very interested in knowing the total size of this batch (that your competitor released to the market) but the competitor keeps the numbers secret.

Fortunately for you, your competitor did not study in an MSBA program and they made a big mistake:

- they stamped each gadget with a consecutive serial number starting with 1.

Your analysts went around the country and (uniformly) randomly purchased a few gadgets from that batch.

These gadgets happen to have serial numbers: 54, 889, 2091, 7612, 20835. From these 5 serial numbers, can you give the best estimate of the *total* number of competitor's gadgets on the market?

This is the famous German tank problem — a classic example of how understanding probabilities can help spot "randomization" in very unexpected situations and use it to solve an important problem.

During World War 2, Germans were well-known for valuing order and thus, were stamping their tanks with serial numbers in a very orderly fashion. However, artillery hits are largely random – artillery destroys tanks independently of their serial number. This allowed Allies to destroy and capture an independent random sample of tanks using artillery as a "randomizer".

Based on the randomness of serial numbers of the destroyed tanks, the clever use of probability allowed the Allies to produce a good estimate of the total count of tanks that Germans had — the invaluable intelligence information.

These estimates proved to be very accurate later once German archives were captured after the end of WW2.

Question 1.

- Adopting a frequentist perspective, please estimate:

  - the (point-estimate of) the total number of gadgets on the market
  - Left-adjusted 95% confidence interval on the total number of gadgets
- Hints:
  - For Q1, feel free to just plugin the formulas.
  - Please use the Minimum-Variance Unbiased Estimator (MVUE) from Wikipedia
- Output:
  - Please create data.frame `df1` with 1 row that contains your point estimate in column `df1$estimate`, lower side of 95% confidence interval `df1$low`, upper side of 95% confidence interval `df1$high`

```
# Please write your code below

k <- 5
m <- 20835

df1 <- data.frame(estimate = round((m*(1+1/k)-1)), low = m, high = round(m*1/(0.05^(1/k))))
head(df1)
```

```
##   estimate   low  high
## 1    25001 20835 37931
```

Question 2

- Adopting a frequentist perspective, please prove with a Monte-Carlo simulation that:
  - Frequentist MVUE from Q1 is indeed unbiased
  - Frequentist 95% confidence interval from Q1 does indeed contain the true value in approximately 95% of the random samples
- Requirements:
  - for the simulation, feel free to assume that:
    - your analysts always purchase 5 items uniformly randomly
    - the true batch size is fixed and is not resampled (please use the size that you obtained in Q1)
- Please create the simulation based on at least $R = 5000L$ samples but makes sure that your Rmd knits in less than 3 minutes.
- Hints:
  - Please use the Minimum-Variance Unbiased Estimator (MVUE) from Wikipedia
- Output:
  - Please create data.frame `df2` with R rows that contains your point estimates in column `df2$estimate`, lower sides of 95% confidence intervals `df2$low`, upper sides of 95% confidence intervals `df2$high`

```
R <- 5000L

# Please write your code below

set.seed(123456)

N <- df1$estimate

df2 <- data.frame(id = rep(1:R,each=k),
                  sp_data = sample(1:25001,k*R,replace=FALSE)) %>%
  group_by(id) %>%
  summarise(m = max(sp_data),
            estimate = round((m*(1+1/k)-1)),
            low = m,
            high = round(m*20^(1/k))) %>%
  select(-m)

head(df2)
```

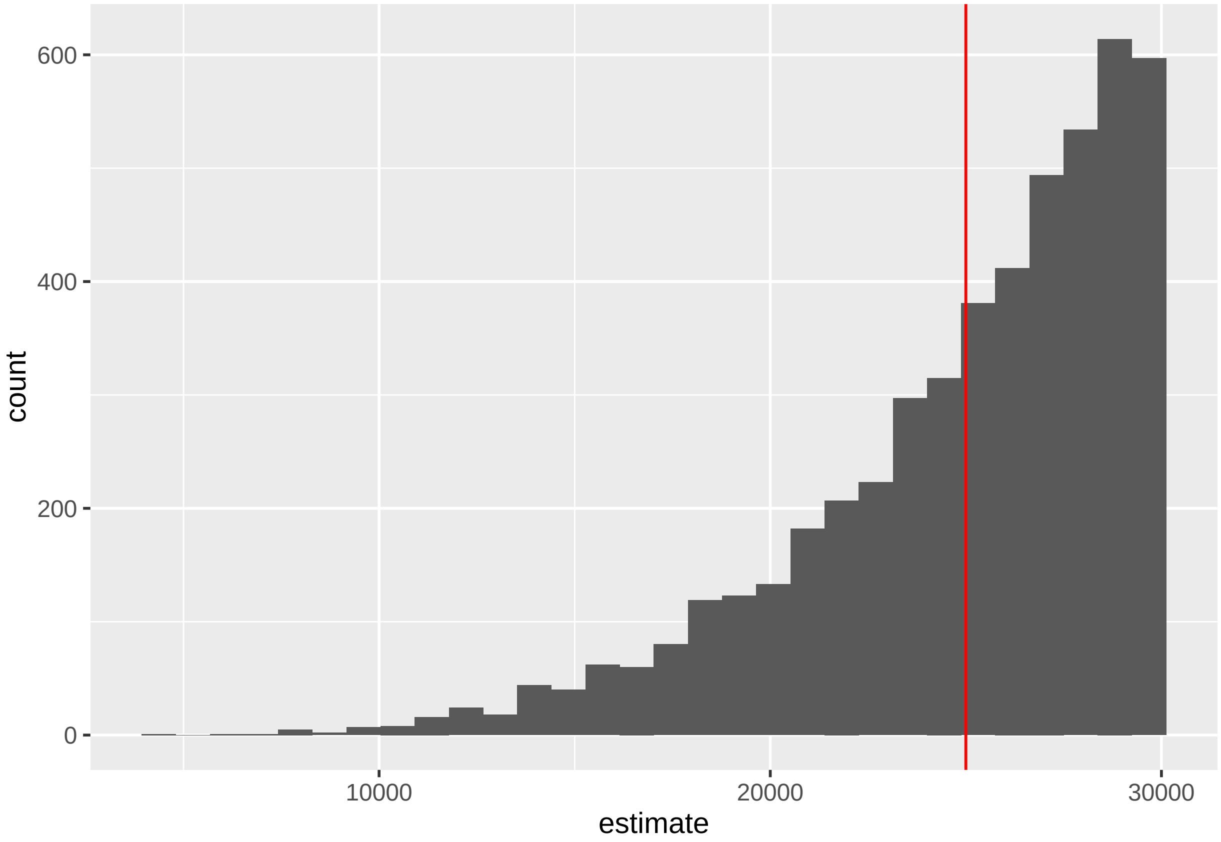
```
## # A tibble: 6 x 4
##      id estimate   low  high
##   <int>    <dbl> <int> <dbl>
## 1     1    23934 19946 36313
## 2     2    29627 24690 44950
## 3     3    29799 24833 45210
## 4     4    26863 22387 40757
## 5     5    14320 11934 21727
## 6     6    26370 21976 40009
```

```
#mean estimate
mean(df2$estimate)
```

```
## [1] 24994.7
```

```
#The mean of estimate is 24994.7, which is very close to the true value of N, that is, 25001,
#so the MVUE is indeed unbiased.

p2 <- ggplot(df2) +
  geom_histogram(aes(estimate), bins=30) +
  geom_vline(aes(xintercept = N),
             color = "red")
p2
```



```
#95% confidence interval
df2 %>%
  mutate(contains = ifelse(low <= N &
                             N <= high,
                           1L,0L)) %>%
  summarise(mean(contains))
```

```
## # A tibble: 1 x 1
##   mean(contains)
##            <dbl>
## 1         0.9506
```

```
#95% confidence intervals indeed contain the true value in approximately 95% of the random samples.
```

Question 3

- Please manually compute Maximum Likelihood Estimator for this problem.
- Please describe:
  - What is the Maximum Likelihood Estimator here?
  - Would you use the estimate that it provides?
- Please remember this example when you hear someone using a "Maximum Likelihood Estimator" and you are dealing with an unfamiliar probabilistic model.
- NOTE: For this problem, please report your computation. At the very least, you need to carefully explain why the number that you found is MLE as described in the assignment. Feel free to consult

Wikipedia but explain everything in your own words

Please write down your answer below: The Maximum Likelihood Estimator is the largest serial number observed, which is 20835 in our case. Computation: We need to find a value for the total number of gadgets(N) so that the likelihood of observing/capturing these 5 serial numbers is as large as it can be. Here we set X=serial number of randomly captured gadgets. So we try to maximize P(X=54,X=889,X=2091,X=7612,X=20835). Here we assume those gadgets that are captured are independent. So P(X=54,X=889,X=2091,X=7612,X=20835) = P(X=54)P(X=889)P(X=2091)P(X=7612)P(X=20835). Since X is uniformly distributed on {1,2,. . . ,N}, so the probability of capturing each gadget is 1/N. So P(X=54,X=889,X=2091,X=7612,X=20835) = P(X=54)P(X=889)P(X=2091)P(X=7612)P(X=20835) = (1/N)(1/N)(1/N)(1/N)(1/N)=1/(Nˆ5). Since N is positive, so in order to maximize 1/(Nˆ5), N has to be as small as possible. However, N can not be smaller than each serial number that we have observed, so the minimun number for N is the largest serial number we observed. In other words,the largest serial number observed is the Maximum Likelihood Estimator for N. We won't use the estimate from Maximum Likelihood Estimator because it is a biased estimator. It will always be less than or equal to the true number of gadgets.