Take Home Exam

Colin Pi

March 13th 2019

Problem 1

```
guat <- read.csv("http://math.carleton.edu/Chihara/Stats275/Guat.csv")
```

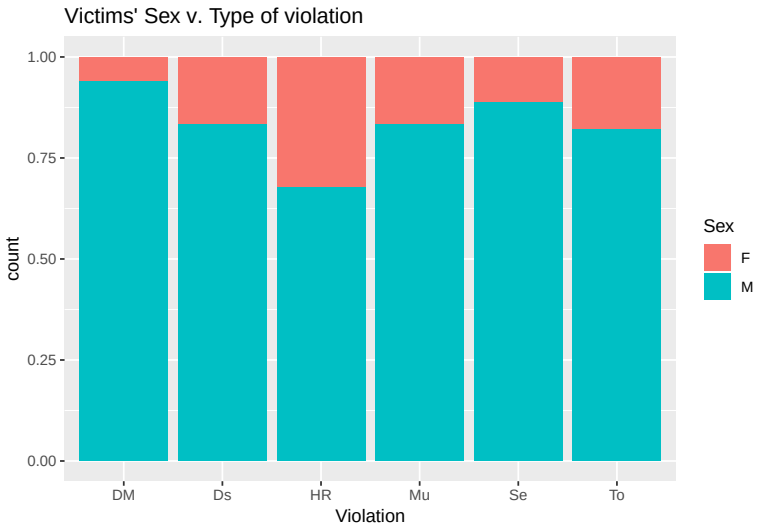
(a)

I will take χ^2 test of independence to see the relationship between sex of the victims and type of violation.

$$H_0 : Types\ of\ violation\ are\ independent\ from\ sex\ of\ the\ victims.$$

$$H_\alpha : Types\ of\ violation\ are\ not\ independent\ from\ sex\ of\ the\ victims.$$

```
ggplot(guat) +
  geom_bar(aes(x = Violation, fill = Sex), position="fill") +
  ggtitle("Victims' Sex v. Type of violation")
```



This graph indicates the victims' sex distribution of each type of violation. HR (injured or army attack) appears to have more female victims relative to male victims compared to other types of crimes as oppose to DM (disappeared, later found killed).

```
table.1 <- table(guat$Sex, guat$Violation)
knitr::kable(table.1, caption = "Contingency Table of the Gender and Types of Crimes")
```

Table 1: Contingency Table of the Gender and Types of Crimes

	DM	Ds	HR	Mu	Se	To
F	2	25	9	161	29	8
M	32	126	19	815	232	37

```r
expect.1 <- round(outer(rowSums(table.1), colSums(table.1))/sum(table.1), digits = 2)
knitr::kable(expect.1, caption = "Expected Counts of the Gender and Types of Crimes")
```

Table 2: Expected Counts of the Gender and Types of Crimes

	DM	Ds	HR	Mu	Se	To
F	5.32	23.63	4.38	152.77	40.85	7.04
M	28.68	127.37	23.62	823.23	220.15	37.96

```r
stat.1 <- sum((table.1 - expect.1)^2/expect.1)
```

We got two cells whose expected counts are less or around 5 (Female, HR = 4.38; Female, DM = 5.32). So the chi-square test statistic (13.076) may not follow χ^2 distribution with degrees of freedom = 5. In that we have a row data, I will use χ^2 permutation test.

```r
set.seed(8)
chisq.test(table.1, simulate.p.value = TRUE, B = 10^5 - 1)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 99999
##  replicates)
##
## data:  table.1
## X-squared = 13.076, df = NA, p-value = 0.02307
```

The χ^2 permutation test produces a p-value of 0.02307, which also provides an enough evidence to reject the claim that the types of violation and sex of the victims are independent.

(b)

```r
male <- subset(guat, select = Age, subset = Sex == "M", drop = T)
female <- subset(guat, select = Age, subset = Sex == "F", drop = T)

stargazer(data.frame(male), header = FALSE, title = "Summary Statistics (Male Age)")
```

Table 3: Summary Statistics (Male Age)

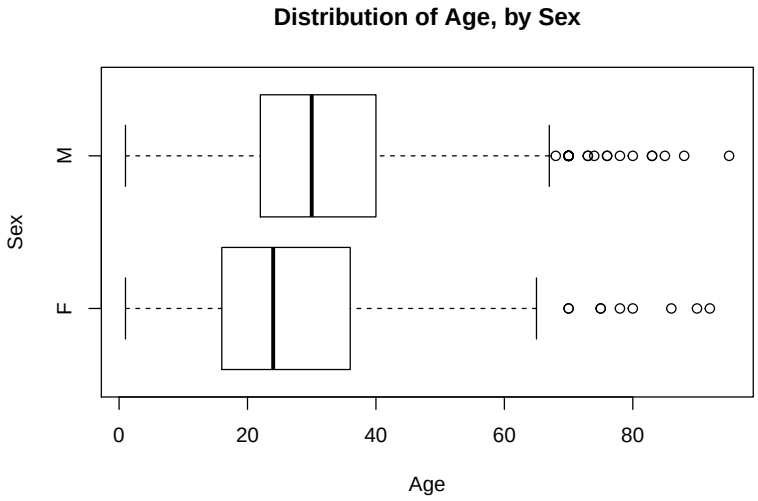
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
male	1,261	32.358	14.210	1	22	40	95

```r
stargazer(data.frame(female), header = FALSE, title = "Summary Statistics (Female Age)")
```

Table 4: Summary Statistics (Female Age)

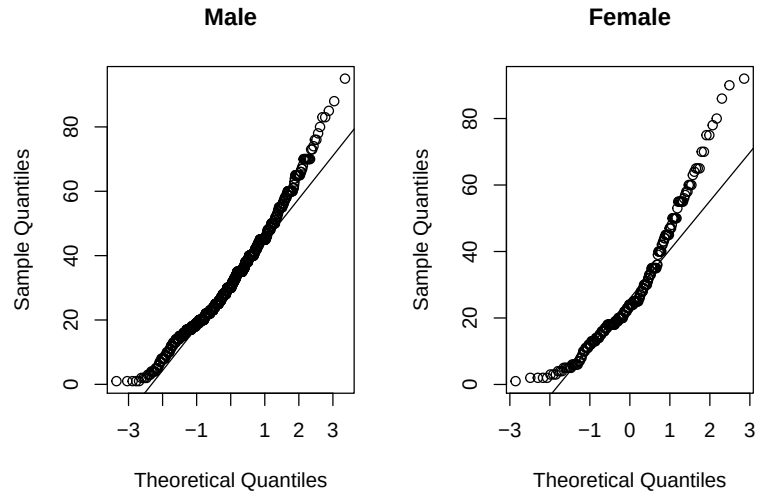
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
female	234	27.966	18.381	1	16	35.8	92

```r
plot(Age ~ Sex, data = guat, main = "Distribution of Age, by Sex", horizontal = TRUE)
```



The observed mean difference in ages of male and female victimes is 4.392634 years. The range of the female victims' age is slightly narrower than that of the male victims' age.

```
par(mfrow=c(1,2))
qqnorm(male, main = "Male")
qqline(male)
qqnorm(female, main = "Female")
qqline(female)
```



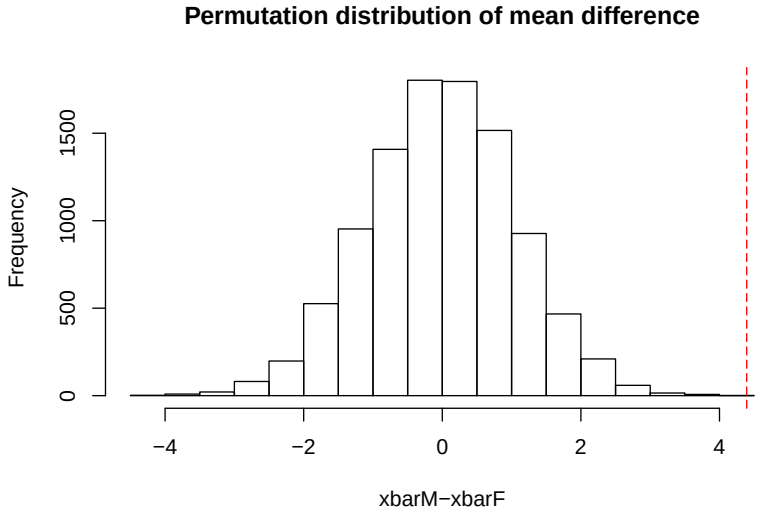

```
par(mfrow=c(1,1))
```

Both boxplot and QQ-plot suggests that the distributions of the age of both male and female victims are skewed to the right. The number of observations in each sex is not balanced (Male = 1,261 observations, Female = 234 observations), so formula t is not appropriate for testing the significance as well as getting the confidence interval. To address the skewness, I will use permutation test for testing out the significance and bootstrap t method to obtain the 95% confidence interval.

$$H_0 : \mu_{male} = \mu_{female}$$
$$H_\alpha : \mu_{male} \neq \mu_{female}$$

where μ is mean of the age.

```r
set.seed(8)

N <- 10^4 - 1
n <- nrow(guat)

mean.diff <- mean(male) - mean(female)
se <- sqrt(var(male)/length(male) + var(female)/length(female))

perm.result <- numeric(N)
Tstar <- numeric(N)
boot.diff <- numeric(N)

for (i in 1:N) {
    ## Permutation Test
    index <- sample(n, size = length(male), replace = FALSE)
    perm.result[i] <- mean(guat$Age[index]) - mean(guat$Age[-index])

    ## Bootstrap T CI
    maleBoot <- sample(male, length(male), replace = TRUE)
    femaleBoot <- sample(female, length(female), replace = TRUE)
    SEstar <- sqrt(var(maleBoot)/length(male) + var(femaleBoot)/length(female))
    meanDiffBoot <- mean(maleBoot) - mean(femaleBoot)
    Tstar[i] <- (meanDiffBoot - mean.diff)/SEstar
}

hist(perm.result, xlab = "xbarM-xbarF", main = "Permutation distribution of mean difference",
    xlim = range(-4.5:4.5))
abline(v = mean.diff, col = "red", lty = 5)
```




```r
p.1.2 <- 2 * (sum(perm.result >= mean.diff) + 1)/(N + 1)  #Two-sided test
p.1.2
```

```
## [1] 2e-04
```

```r
boot.t.CI <- mean.diff - quantile(Tstar, c(0.975, 0.025)) * se
boot.t.CI
```

```
##    97.5%     2.5%
```

```
## 1.771229 6.756739
```

We obtained a p-value of 0.00019998, which provides an enough evidence to reject the claim that the mean age of male and female victims are the same. The 95% confidence interval tells us that we are 95% confident that in average the age of male victims is 1.771 to 6.757 years older than the age of female victims.

Problem 2

To investigate the how robust this confidence interval when the sample is drawn from the non-normal distributions, I'm going to estimate the 95% CIs of the variance of the sample drawn from the four different distributions (Normal, Exponential, Chi-square, T) and compare how well these approximated CIs capture the true variance among the different distributions. I will also tweak the population variance, degrees of freedom (for t distribution), and the sample size to see whether the result significantly differs by these factors.

- Variance of χ^2 distribution: $2 \cdot df$
- Variance of Exponential distribution: $\frac{1}{\lambda^2}$
- Variance of T distribution: $\frac{df}{df-2}$

```
tester <- function(dist, n, variance, tdf) {
  N <- 10^4
  tooLow <- 0
  tooHigh <- 0
  for (i in 1:N){
    if (dist == "Normal") x <- rnorm(n,0,sqrt(variance))
    if (dist == "Chisq") x <- rchisq(n,variance/2)
    if (dist == "Exp") x <- rexp(n,1/sqrt(variance))
    if (dist == "T") {
      x <- rt(n,tdf)
      variance <- tdf/(tdf-2)
      ## for t dist, getting the variance from df is more straightforward than the other way around
    }
    L <- (n-1)*var(x)/qchisq(0.975,n-1)
    U <- (n-1)*var(x)/qchisq(0.025,n-1)
    if (L > variance)
      tooHigh <- tooHigh + 1 ## true variance below CI, increase the counter
    if (U < variance)
      tooLow <- tooLow + 1 ## true variance above CI, increase the counter
  }

  results <- c(tooLow, tooHigh)/N
  return(results)
}
```

Normal Distribution

Var = 4

```
set.seed(8)
tester("Normal",100,4,0)
```

```
## [1] 0.0246 0.0250
```

Var = 36

```
set.seed(8)
tester("Normal",100,36,0)
```

```
## [1] 0.0246 0.0250
```

Var = 400

```
set.seed(8)
tester("Normal",100,400,0)
```

```
## [1] 0.0246 0.0250
```

Regardless of the change in variance, the 95% of approximated CIs capture the true variance. Around 2.5% of the approximated CIs are lower than the true variance, and 2.5% of the approximated CIS are above the true variance (symmetric).

N = 1000

Var = 4

```
set.seed(8)
tester("Normal",1000,4,0)
```

```
## [1] 0.0252 0.0260
```

Var = 36

```
set.seed(8)
tester("Normal",1000,36,0)
```

```
## [1] 0.0252 0.0260
```

Var = 400

```
set.seed(8)
tester("Normal",1000,400,0)
```

```
## [1] 0.0252 0.0260
```

Increasing the sample size does not create a huge difference in results from the cases of a smaller sample size. The performance got slightly worsened.

Exponential Distribution

Var = 4

```
set.seed(8)
tester("Exp",100,4,0)
```

```
## [1] 0.1678 0.1375
```

Var = 36

```
set.seed(8)
tester("Exp",100,36,0)
```

```
## [1] 0.1678 0.1375
```

Var = 400

```r
set.seed(8)
tester("Exp",100,400,0)
```

```
## [1] 0.1678 0.1375
```

Regardless of the change in variance, the 70% of approximated CIs capture the true variance (way lower than 95%). Around 16% of the approximated CIs are lower than the true variance, and 14% of the approximated CIs are above the true variance (not symmetric).

N = 1000

Var = 4

```r
set.seed(8)
tester("Exp",1000,4,0)
```

```
## [1] 0.1645 0.1601
```

Var = 36

```r
set.seed(8)
tester("Exp",1000,36,0)
```

```
## [1] 0.1645 0.1601
```

Var = 400

```r
set.seed(8)
tester("Exp",1000,400,0)
```

```
## [1] 0.1645 0.1601
```

The approximated CIs performs worse in a larger sample size. However, the times that the CIs miss the true variance in either side (either higher or lower than the true variance) got more symmetric (around 16%).

Chi-square Distribution

Var = 4

```r
set.seed(8)

tester("Chisq",100,4,0)
```

```
## [1] 0.1757 0.1426
```

The 70% of approximated CIs capture the true variance (way lower than 95%). Around 17% of the approximated CIs are lower than the true variance, and 14% of the approximated CIS are above the true variance (not symmetric).

Var = 36

```r
set.seed(8)
tester("Chisq",100,36,0)
```

```
## [1] 0.0406 0.0442
```

The 90% of approximated CIs capture the true variance (slightly lower than 95%). Around 4% of the approximated CIs are lower than the true variance, and 4.4% of the approximated CIs are above the true variance (approximately symmetric).

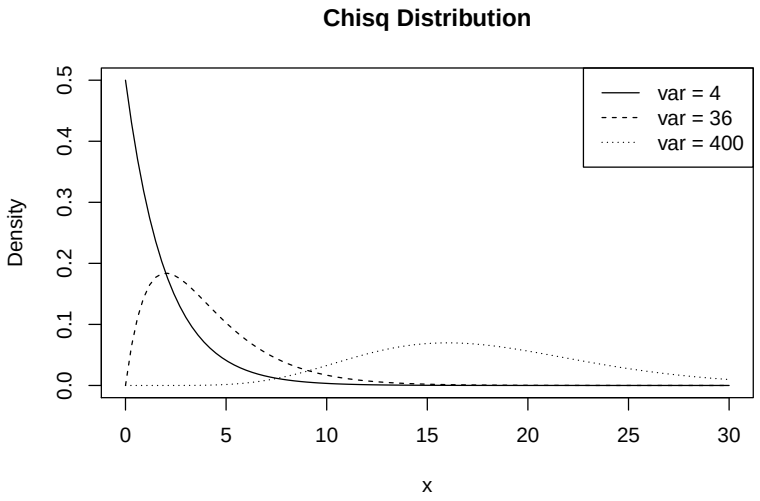
Var = 400

```
set.seed(8)
tester("Chisq",100,400,0)
```

```
## [1] 0.0254 0.0285
```

The 95% of approximated CIs capture the true variance. Around 2.5% of the approximated CIs are lower than the true variance, and 2.8% of the approximated CIS are above the true variance (approximately symmetric).

```
curve(dchisq(x,2), from = 0, to = 30, ylab = "Density", main = "Chisq Distribution")
curve(dchisq(x,4), from = 0, to = 30, add = TRUE, lty = 2)
curve(dchisq(x,18), from = 0, to = 30, add = TRUE, lty = 3)
legend("topright", legend = c("var = 4","var = 36","var = 400"), lty = 1:3)
```



As illustrated above, χ^2 distribution becomes more normal as variance increases. We might assume that's why the approximated CIs perform better as variance increases.

N = 1000

Var = 4

```
set.seed(8)
tester("Chisq",1000,4,0)
```

```
## [1] 0.1707 0.1562
```

Var = 36

```
set.seed(8)
tester("Chisq",1000,36,0)
```

```
## [1] 0.0458 0.0489
```

Var = 400

```
set.seed(8)
tester("Chisq",1000,400,0)
```

```
## [1] 0.0256 0.0271
```

Increasing the sample size does not create a huge difference in results from smaller sample size.

T Distribution

Df = 4

```
set.seed(8)
tester("T",100,0,4)
```

```
## [1] 0.1771 0.1232
```

The 70% of approximated CIs capture the true variance (way lower than 95%). Around 17% of the approximated CIs are lower than the true variance, and 12% of the approximated CIS are above the true variance (not symmetric).

Df = 40

```
set.seed(8)
tester("T",100,0,40)
```

```
## [1] 0.0292 0.0302
```

The 94% of approximated CIs capture the true variance (approximately 95%). Around 3% of the approximated CIs are lower than the true variance, and 3% of the approximated CIS are above the true variance (symmetric).

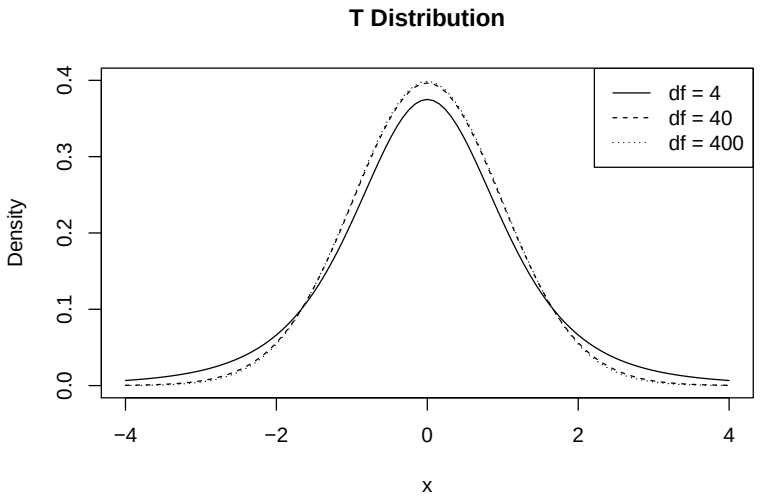
Df = 400

```
set.seed(8)
tester("T",100,0,400)
```

```
## [1] 0.026 0.027
```

The 95% of approximated CIs capture the true variance. Around 2.6% of the approximated CIs are lower than the true variance, and 2.7% of the approximated CIs are above the true variance (symmetric).

```
curve(dt(x,4), from = -4, to = 4, ylim = c(0,0.4), ylab = "Density", main = "T Distribution")
curve(dt(x,40), from = -4, to = 4, add = TRUE, lty = 2)
curve(dt(x,400), from = -4, to = 4, add = TRUE, lty = 3)
legend("topright", legend = c("df = 4","df = 40","df = 400"), lty = 1:3)
```



As illustrated above, t distribution becomes more normal as df increases. We might assume that's why the approximated CIs perform better as df increases.

N = 1000

Df = 4

```
set.seed(8)
tester("T",1000,0,4)
```

```
## [1] 0.2317 0.1651
```

Df = 40

```
set.seed(8)
tester("T",1000,0,40)
```

```
## [1] 0.0314 0.0282
```

Df = 400

```
set.seed(8)
tester("T",1000,0,400)
```

```
## [1] 0.0259 0.0249
```

Increasing the sample size does not create a huge difference in results from smaller sample size.

Conclusion

Overall, $((n-1)S^2/q_2, (n-1)S^2/q_1)$ only works well when the sample is drawn from normal distribution but works poorly most of the time when the sample is drawn from population that is not normally distributed. We can observe that the approximated CIs perform better as the population distribution gets closer to normal distribution (by increasing variance for χ^2 and df for t), a phenomenon that also supports the claim that $((n-1)S^2/q_2, (n-1)S^2/q_1)$ only works well under normality assumption.