MAST20005/MAST90058: Week 5 Lab Solutions

1. There are many possible approximations you can use. Here are two of the default choices:

```
binom.test(24, 642)

##
##  Exact binomial test
##
## data:  24 and 642
## number of successes = 24, number of trials = 642, p-value <
## 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.02409629 0.05511449
## sample estimates:
## probability of success
##             0.03738318

prop.test(24, 642)

##
##  1-sample proportions test with continuity correction
##
## data:  24 out of 642, null probability 0.5
## X-squared = 547.74, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.02461784 0.05593801
## sample estimates:
##          p
## 0.03738318
```

And similarly for the one-sided CI:

```
binom.test(24, 642, alternative = "less")

##
##  Exact binomial test
##
## data:  24 and 642
## number of successes = 24, number of trials = 642, p-value <
## 2.2e-16
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.00000000 0.05217303
## sample estimates:
## probability of success
##             0.03738318
```

```r
prop.test(24, 642, alternative = "less")

##
##  1-sample proportions test with continuity correction
##
## data:  24 out of 642, null probability 0.5
## X-squared = 547.74, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
##  0.00000000 0.05266172
## sample estimates:
##          p
## 0.03738318
```

So we can summarise our answers as:

(a) **0.037**

(b) (**0.024**, **0.055**)

(c) Upper bound: **0.052**

2. (a) Let's use the the built-in `t.test()` function. Unfortunately, we haven't been given the raw data. However, there's a nice trick we can use here. Since we know that the only information that will be used to construct the CI are the statistics $\bar{x}$ and s, and the sample size n, if we construct a synthetic dataset with the same values of these quantities then we will get the same CI as if we used the real data!

```r
# Generate some raw values to work with.
x <- rnorm(10)
y <- rnorm(10)

# Shift and scale to match the given statistics.
x <- (x - mean(x)) / sd(x)  # standardise
y <- (y - mean(y)) / sd(y)
x <- 2.548 + 0.323 * x      # 'unstandardise'
y <- 1.564 + 0.210 * y

# Now do the inference.
t.test(x, y, var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  x and y
## t = 8.0767, df = 18, p-value = 2.139e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.7280416 1.2399584
## sample estimates:
## mean of x mean of y
##     2.548     1.564
```

(b) (See tutorial solutions)

(c) We can compute a CI for the ratio of variances:

```
var.test(x, y)

##
##  F test to compare two variances
##
## data:  x and y
## F = 2.3657, num df = 9, denom df = 9, p-value = 0.2156
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.5876156 9.5244432
## sample estimates:
## ratio of variances
##           2.365737
```

3.
```
x <- c(33.8, 32.2, 30.7, 35.4, 31, 30.3, 26.8, 33.2, 27.8, 27.2)
n <- length(x)
```

(a)
```
mean(x)  # point estimate

## [1] 30.84

mean(x) + c(-1, 1) * qt(0.95, n - 1) * sd(x) / sqrt(n)  # 90% CI

## [1] 29.15411 32.52589
```

(b)
```
sd(x)  # point estimate

## [1] 2.908302

sqrt((n - 1) / qchisq(c(0.975, 0.025), df = n - 1)) * sd(x)  # 95% CI

## [1] 2.000433 5.309426
```

(c)
```
mean(x) + c(-1, 1) * qt(0.95, n - 1) * sd(x) * sqrt(1 + 1 / n)  # 90% PI

## [1] 25.24854 36.43146
```

4.
```
B <- 10000
inside <- logical(B)
for (i in 1:B) {
    x <- rnorm(100, 10, 2)
    ci <- mean(x) + c(-1, 1) * qnorm(0.975) * sd(x) / sqrt(100)
    inside[i] <- ci[1] < 10 & 10 < ci[2]  # check if true value inside CI
}
mean(inside)  # estimate of the coverage (should be close to 95%)

## [1] 0.9483
```