Chapter 7

Monte Carlo Methods in Inference

Packages needed for these exercises: ggplot2.

7.1 *Estimate the MSE of the level k trimmed means for random samples of size 20 generated from a standard Cauchy distribution. (The target parameter θ is the center or median; the expected value does not exist.) Summarize the estimates of MSE in a table for $k = 1, 2, \ldots, 9$.*

```
n <- 20
K <- n/2 - 1
m <- 1000
mse <- matrix(0, n/2, 2)

trimmed.mse <- function(n, m, k) {
    tmean <- numeric(m)
    for (i in 1:m) {
        x <- sort(rcauchy(n))
        tmean[i] <- sum(x[(k+1):(n-k)]) / (n-2*k)
        }
    mse.est <- mean(tmean^2)
    se.est <- sqrt(mean((tmean-mean(tmean))^2)) / sqrt(m)
    return(c(mse.est, se.est))
}

for (k in 0:K)
    mse[k+1, 1:2] <- trimmed.mse(n=n, m=m, k=k)

mse <- as.data.frame(cbind(0:K, mse))
```

```r
    names(mse) <- list("k", "t-mean", "se")
    print(mse)
```

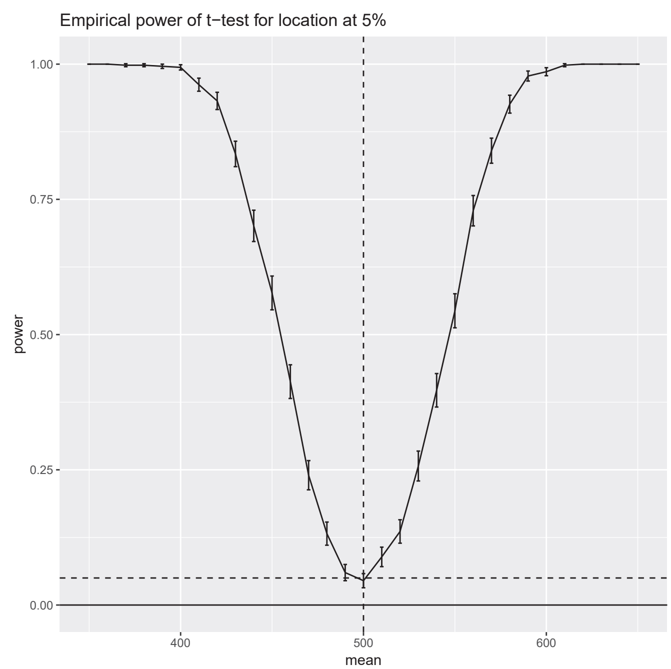
```
##    k     t-mean         se
## 1  0 82.6578971 0.28749176
## 2  1  1.3422777 0.03663287
## 3  2  0.3478463 0.01865002
## 4  3  0.2516950 0.01584669
## 5  4  0.1708088 0.01305890
## 6  5  0.1503993 0.01226187
## 7  6  0.1441091 0.01199713
## 8  7  0.1352604 0.01161845
## 9  8  0.1352857 0.01162528
## 10 9  0.1400206 0.01183221
```

7.2 *Plot the empirical power curve for the t-test, changing the alternative hypothesis to $H_1 : \mu \neq 500$, and keeping the significance level $\alpha = 0.05$.*

```r
    n <- 20
    m <- 1000
    mu0 <- 500
    sigma <- 100
    mu <- c(seq(350, 650, 10))  #alternatives
    M <- length(mu)
    power <- numeric(M)
    for (i in 1:M) {
        mu1 <- mu[i]
        pvalues <- replicate(m, expr = {
            #simulate under alternative mu1
            x <- rnorm(n, mean = mu1, sd = sigma)
            ttest <- t.test(x,
                     alternative = "two.sided", mu = mu0)
            ttest$p.value  } )
        power[i] <- mean(pvalues <= .05)
    }
    se <- sqrt(power * (1-power) / m)
```

```r
suppressMessages(library(ggplot2))
dat <- data.frame(mean=mu, power=power,
                  upper=power+2*se, lower=power-2*se)
ggplot(dat, aes(x=mean, y=power)) +
    geom_line() +
    geom_hline(yintercept=c(0,0.05), lty=1:2) +
```

```r
geom_vline(xintercept=mu0, lty=2) +
geom_errorbar(aes(ymin=lower, ymax=upper), width=2) +
ggtitle("Empirical power of t-test for location at 5%")
```



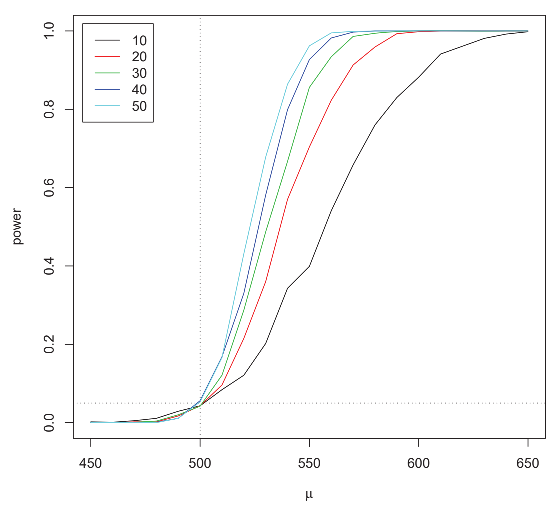
7.3 *Plot the power curves for the one-sided t-test for sample sizes 10, 20, 30, 40, and 50, but omit the standard error bars. Plot the curves on the same graph, each in a different color or different line type, and include a legend.*

```r
N <- c(10, 20, 30, 40, 50)
m <- 1000
mu0 <- 500
sigma <- 100
mu <- c(seq(450, 650, 10))  #alternatives
M <- length(mu)
power <- matrix(0, M, 5)

for (j in 1:5) {
    n <- N[j]
    for (i in 1:M) {
        mu1 <- mu[i]
        pvalues <- replicate(m, expr = {
            #simulate under alternative mu1
```

```
            x <- rnorm(n, mean = mu1, sd = sigma)
            ttest <- t.test(x,
                    alternative = "greater", mu = mu0)
            ttest$p.value  } )
        power[i, j] <- mean(pvalues <= .05)
    }
}
```

```
plot(mu, power[, 1], type="l", ylim=range(power),
     xlab=bquote(mu), ylab="power")
abline(v = mu0, lty = 3)
abline(h = .05, lty = 3)
for (j in 2:5)
    lines(mu, power[ ,j], col=j)
legend("topleft", inset=.02, legend=N, col=1:5, lty=1)
```

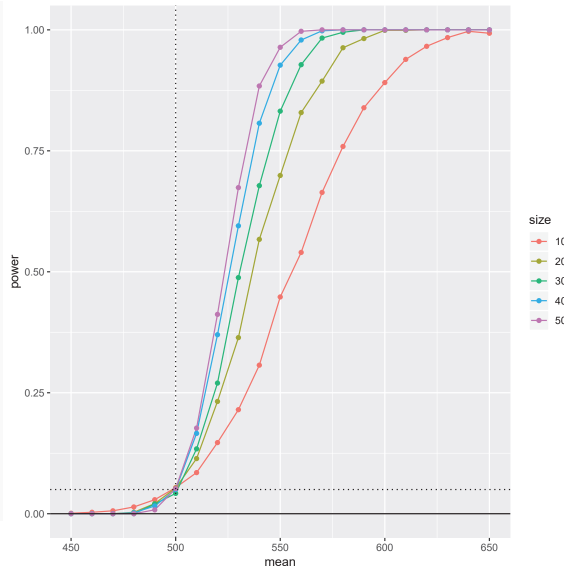


The plots show that for a fixed alternative, the power is increasing with sample size.

To produce a similar plot using ggplot, the estimates should be stored in a data frame with a variable that identifies the simulation (the sample size).

```
pwr <- data.frame(n=rep(N, each=length(mu)),
                  mean=rep(mu, times=length(N)),
                  power=numeric(length(N)*length(mu)))
```

```r
for (i in 1:nrow(pwr)) {
    n <- pwr$n[i]
    mu1 <- pwr$mean[i]
    pvalues <- replicate(m, expr = {
        #simulate under alternative mu1
        x <- rnorm(n, mean = mu1, sd = sigma)
        ttest <- t.test(x,
                    alternative = "greater", mu = mu0)
        ttest$p.value  } )
    pwr$power[i] <- mean(pvalues <= .05)
}
pwr$size <- as.factor(pwr$n)
ggplot2::ggplot(pwr, aes(mean, power, color=size)) +
    geom_line() + geom_point() +
    geom_hline(yintercept=c(0,0.05), lty=c(1,3)) +
    geom_vline(xintercept=500, lty=3)
```



7.5 *Refer to Example 1.6 (run length encoding). Use simulation to estimate the probability that the observed maximum run length for the fair coin flipping experiment is in [9,11] in a sample size of 1000. Use the results of your simulation to estimate the standard error of the maximum run length for this experiment. Suppose that you observed 1000 coin flips and the maximum run length was 9. Would you suspect that the coin is unfair? Explain.*

```r
    n <- 1000
    out <- replicate(m, expr={
        x <- rbinom(n, size=1, prob=.5)
        r <- rle(x)
        runlengths <- r$lengths
        max(runlengths)
    })
    summary(out)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00    9.00   10.00   10.28   11.00   21.00

    sd(out)

## [1] 1.884921
```

```r
    p <- 1 - mean(out < 9 | out > 11)
    p
```

```
## [1] 0.65
```

```r
    cdf9 <- mean(out <= 9)
    cdf9
```

```
## [1] 0.384
```

The standard error of the maximum RLE is approximately 1.8849212. The probability that $9 \leq \max RLE \leq 11$ is approximately 0.65. The probability of observing max RLE 9 or less is approximately 0.384, which is not unusual for a fair coin and does not support a hypothesis that the coin is unfair.

7.6 *Use a Monte Carlo experiment to estimate the coverage probability of the t-interval for random samples of $\chi^2(2)$ data with sample size $n = 20$. Compare your t-interval results with the simulation results in Example 7.4.*

```r
    n <- 20
    rootn <- sqrt(n)
    t0 <- qt(c(.025,.975), df=n-1)
    CI <- replicate(10000, expr = {
        x <- rchisq(n, df=2)
            ci <- mean(x) + t0 * sd(x) / rootn
        } )
    LCL <- CI[1,]
    UCL <- CI[2,]
    sum(LCL < 2 & UCL > 2)
```

```
## [1] 9140
```

```r
    mean(LCL < 2 & UCL > 2)
```

```
## [1] 0.914
```

The achieved confidence level is 0.914, which is lower than the nominal level 0.95, but has better coverage than in Example 7.4. For the $\chi^2(2)$ distribution the empirical coverage rate was only 77.3%. The t-interval is more robust to departures from normality than the interval for variance.

7.7 *Estimate the 0.025, 0.05, 0.95 and 0.975 quantiles of the skewness $\sqrt{b_1}$ under normality by a Monte Carlo experiment. Compute the standard error of the estimates using the normal approximation for the density (with exact*

variance formula). Compare the estimated quantiles with the quantiles of the large sample approximation $\sqrt{b_1} \approx N(0, 6/n)$.
Equation (2.14) gives the variance of a sample quantile:

$$Var(\hat{x}_q) = \frac{q(1-q)}{nf(x_q)^2}.$$

Here the density f is the density of the skewness statistic, and the value of n is the number of replicates of the statistic. To estimate $se(\hat{x}_q)$, we are approximating f with the asymptotic normal density.

```
    sk <- function(x) {
        #computes the sample skewness coeff.
        xbar <- mean(x)
        m3 <- mean((x - xbar)^3)
        m2 <- mean((x - xbar)^2)
        return( m3 / m2^1.5 )
    }

    m <- 10000                      #num. repl. each sim.
    n <- 50

    skstats <- replicate(m, expr = {
            x <- rnorm(n)
            sk(x) })

    p <- c(.025, .05, .95, .975)
    q1 <- quantile(skstats, p)
    q2 <- qnorm(p, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))
    q3 <- qnorm(p, 0, sqrt(6 / n))
    f <- dnorm(q2, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))
    v <- p * (1 - p) / (m * f^2)
    rbind(p, q1, se=sqrt(v))

##              2.5%          5%        95%       97.5%
## p    0.025000000  0.05000000 0.95000000 0.975000000
## q1 -0.624608384 -0.52408120 0.53089163 0.649662701
## se   0.008719622  0.00689781 0.00689781 0.008719622

    rbind(q1, q2, q3)

##           2.5%         5%       95%     97.5%
## q1 -0.6246084 -0.5240812 0.5308916 0.6496627
## q2 -0.6397662 -0.5369087 0.5369087 0.6397662
## q3 -0.6789514 -0.5697940 0.5697940 0.6789514
```

The first table shows the sample quantiles of the skewness statistic and standard error of the estimate for sample size 50.

The second table shows the three estimates of quantiles q_1 (sample) q_2 (normal with exact variance) and q_3 normal asymptotic distribution. The estimated quantiles q_2 are closer to the empirical quantiles than the estimates q_3 using the asymptotic variance $6/n$.

7.8 *Estimate the power of the skewness test of normality against symmetric Beta(α, α) distributions and comment on the results.*

```
alpha <- .1
n <- 30
m <- 2500
ab <- 1:10
N <- length(ab)
pwr <- numeric(N)
#critical value for the skewness test
cv <- qnorm(1-alpha/2, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))

for (j in 1:N) {
    a <- ab[j]
    sktests <- numeric(m)
    for (i in 1:m) {        #for each replicate
        x <- rbeta(n, a, a)
        sktests[i] <- as.integer(abs(sk(x)) >= cv)
        }
    pwr[j] <- mean(sktests)
    }
pwr

##  [1] 0.0180 0.0196 0.0312 0.0344 0.0420 0.0472 0.0608 0.0580 0.0532 0.0588
```

The symmetric beta alternatives are not normal, but symmetric. This simulation illustrates that the skewness test of normality is not very effective against light-tailed symmetric alternatives. The empirical power of the test is not higher than the nominal significance level.

Are the results different for heavy-tailed symmetric alternatives such as $t(\nu)$? Yes, the skewness test is more effective against a heavy-tailed symmetric alternative, such as a Student t distribution. Below we repeat the simulation for several choices of degrees of freedom.

```r
alpha <- .1
n <- 30
m <- 2500
df <- c(1:5, seq(10,50,10))
N <- length(df)
pwr <- numeric(N)
#critical value for the skewness test
cv <- qnorm(1-alpha/2, 0, sqrt(6*(n-2) / ((n+1)*(n+3))))

for (j in 1:N) {
    nu <- df[j]
    sktests <- numeric(m)
    for (i in 1:m) {       #for each replicate
        x <- rt(n, df=nu)
        sktests[i] <- as.integer(abs(sk(x)) >= cv)
        }
    pwr[j] <- mean(sktests)
    }
data.frame(df, pwr)
```

```
##    df    pwr
## 1   1 0.8680
## 2   2 0.6636
## 3   3 0.4956
## 4   4 0.4096
## 5   5 0.3440
## 6  10 0.2044
## 7  20 0.1340
## 8  30 0.1100
## 9  40 0.1252
## 10 50 0.1208
```

The skewness test of normality is more powerful when the degrees of freedom are small. As degrees of freedom tend to infinity the t distribution tends to normal, and the power tends to α. One reason that the skewness test is more powerful in this case than against the symmetric beta distributions is that $|\sqrt{b_1}|$ is positively correlated with kurtosis. Kurtosis of beta distribution is less than the normal kurtosis, while kurtosis of t is greater than the normal kurtosis.