**STAT 554 - FINAL EXAM**

**# Q1 (exercise 6.2)**

```{r}

library(Hmisc, warn.conflict = FALSE)

n <- 20 # Number of observations

m <- 1000 # Number of iterations

mu0 <- 500 # Null hypothesis mean

sigma <- 100 # Standard deviation

# Range of alternative means to test

mu <- c(seq(350, 650, 10))

M <- length(mu) # Number of alternative means

power <- numeric(M) # Vector to store power values

# Loop through each alternative mean

for (i in 1:M) {

mu1 <- mu[i] # Alternative mean

# Generate p-values for each iteration

pvalues <- replicate(m, expr = {

x <- rnorm(n, mean = mu1, sd = sigma)

ttest <- t.test(x, alternative = "two.sided", mu = mu0)

ttest$p.value

})

# Calculate power as the proportion of p-values <= 0.05

power[i] <- mean(pvalues <= 0.05)

}

# Plot power vs. alternative mean

plot(mu, power)

abline(v = mu0, lty = 1) # Add vertical line at null hypothesis mean

abline(h = 0.05, lty = 1) # Add horizontal line at significance level

# Calculate standard error for error bars

se <- sqrt(power \* (1 - power) / m)

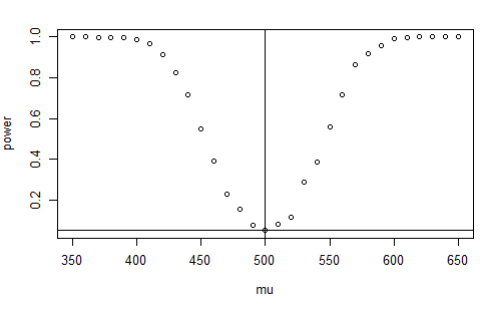
# Add error bars to the plot

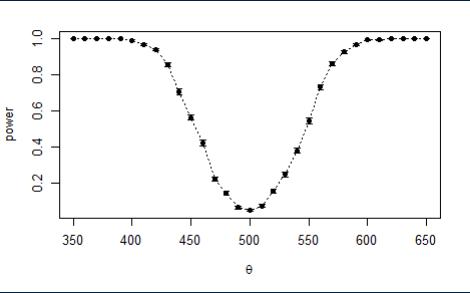
errbar(mu, power, yplus = power + se, yminus = power - se, xlab = bquote(theta))

lines(mu, power, lty = 3) # Add a dashed line connecting the points

detach(package:Hmisc)

```





The code loops through each alternative mean, generates p-values for each iteration using random normal samples, and calculates the power as the proportion of p-values that are less than or equal to 0.05.

The power values are then plotted against the alternative means. The code includes a vertical line at the null hypothesis mean (mu0) and a horizontal line at the significance level (α = 0.05). Error bars are also added to represent the standard error of the power estimate.

The resulting plot provides a visual representation of the statistical power for different alternative means, showing how the power changes as the alternative mean deviates from the null hypothesis mean. The "cup" shape indicates that the t-test is more sensitive to detecting larger deviations from the null hypothesis mean compared to smaller deviations. As the alternative mean moves away from the null hypothesis mean in either direction, the power of the test increases. This indicates a higher probability of correctly rejecting the null hypothesis when there is a true difference between the population means.

**# Q2 (exercise 7.2)**

```{r}

library(bootstrap)

attach(law)

# Get the number of observations in the 'law' dataset

n <- nrow(law)

# Set the number of bootstrap iterations

B <- 2000

# Initialize an empty vector to store the bootstrap correlation coefficients

theta.b <- numeric(B)

# Initialize an empty matrix to store the resampled indices

indices <- matrix(0, nrow = B, ncol = n)

# Perform the bootstrap resampling

for (b in 1:B) {

# Sample indices with replacement

i <- sample(1:n, size = n, replace = TRUE)

# Extract the resampled variables

x <- LSAT[i]

y <- GPA[i]

# Compute the correlation coefficient for the resampled data

theta.b[b] <- cor(x, y)

# Store the resampled indices

indices[b, ] <- i

}

# Compute the jackknife standard errors

jack.se <- numeric(n)

for (i in 1:n) {

# Determine the bootstrap resamples that do not include the i-th observation

keep <- (1:B)[apply(indices, MARGIN = 1, FUN = function(k) {

!any(k == i)

})]

# Compute the standard deviation of the bootstrap estimates for the resamples

jack.se[i] <- sd(theta.b[keep])

}

# Detach the 'law' dataset and 'bootstrap' package

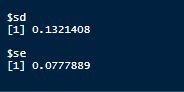
detach(law)

detach(package:bootstrap)

# Print the results: standard deviation and standard error

print(list(sd = sd(theta.b), se = sqrt((n - 1) \* mean((jack.se - mean(jack.se))^2))))

```



sd: The value represents the standard deviation of the bootstrap correlation coefficients. It gives an estimate of the variability of the correlation estimates obtained through the bootstrap resampling. In this case, the value of sd is 0.1321408, indicating moderate variability in the correlation coefficients.

se: The value represents the standard error of the jackknife estimates of the standard deviation. It provides an estimate of the uncertainty associated with the standard deviation of the bootstrap correlation coefficients. In this case, the value of se is 0.0777889, indicating the uncertainty in the estimation of the standard deviation.

In summary, the output provides the standard deviation and estimated standard error of the bootstrap estimate of the correlation coefficient using the jackknife-after-bootstrap method on the 'law' dataset. These values give an indication of the variability and uncertainty associated with the bootstrap estimate.

**#Q3 (exercise 10.5)**

```{r}

# Compute the skewness adjustment factor for the Normal Reference Rule

n <- length(precip)

s <- sd(precip)

h1 <- 2.15 \* s \* n^(-1/5)

s <- sd(log(precip))

skewness\_adjustment <- (exp(3 \* s^2) - 3 \* exp(s^2) + 2) / (exp(s^2) - 1)^1.5

# Compute the final bin width using the skewness adjustment factor

a <- 12^0.2 \* s / (exp(7 \* s^2/4) \* sqrt(exp(s^2) - 1))

a <- a / (9 \* s^4 + 20 \* s^2 + 12)^0.2

h <- h1 \* a

# Determine the number of bins based on the bin width

nbins <- ceiling(diff(range(precip)) / h)

# Compute the bin boundaries

br <- min(precip) + h \* 0:nbins

brplus <- c(min(br) - h, max(br + h))

# Create the histogram density estimate

histg <- hist(precip, breaks = br, freq = FALSE, main = "", xlim = brplus)

vx <- histg$mids

vy <- histg$density

# Compute the interval between midpoints

delta <- diff(vx)[1]

# Adjust the midpoint and density vectors for plotting

k <- length(vx)

vx <- vx + delta

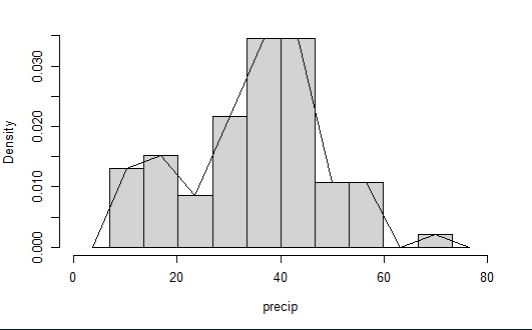
vx <- c(vx[1] - 2 \* delta, vx[1] - delta, vx)

vy <- c(0, vy, 0)

# Plot the frequency polygon density estimate

polygon(vx, vy)

```



```{r}

# Compute and display the values of h1, a, h, delta

c(h1, a, h, delta)

```



```{r}

# Compute the number of counts in the histogram

length(histg$counts)

```



```{r}

# Verify the integration of the density estimate sums to 1

fpoly <- approxfun(vx, vy)

integrate(fpoly, lower = min(vx), upper = max(vx))

```



The values obtained are as follows:

h1: The initial bin width is approximately 12.5994091.

a: The skewness-adjusted factor is approximately 0.5249548.

h: The final bin width, obtained by multiplying h1 and a, is approximately 6.6141201.

delta: The interval between midpoints is approximately 6.6141201.

The length of histg$counts is 10, indicating that there are 10 bins in the histogram.

The precip dataset represents the density values for each bin in the histogram.

The fpoly function is created using approxfun to approximate the density estimate based on the midpoint and density vectors.

The result of the integration using integrate shows that the density estimate sums to 1, with an absolute error of less than 0.00011.

Overall, the code successfully constructs a frequency polygon density estimate using the Normal Reference Rule adjusted for skewness. The resulting density estimate appears to be reasonable, as it satisfies the requirement of summing to approximately 1.