562_HW_2

Matthew Stoebe

2024-10-25

#3 b:

```
x = qchisq(.01, 8)
print(x)
```

```
## [1] 1.646497
```

```
xadj <- x*225 / 100
print(xadj)
```

```
## [1] 3.704619
```

```
pchisq(xadj, df=8, lower.tail=TRUE)
```

```
## [1] 0.1172577
```

#3 c:

```
# Define parameters
threshold <- 0.8          # Desired power
alpha <- 0.01             # Significance level
sigma0_sq <- 225          # Variance under H0
sigma_a_sq <- 100         # Variance under H1

# Starting sample size
n <- 9                    # Corresponds to df = 8

# Initialize variables
df <- n - 1        # Degrees of freedom
power <- 0                 # Initial power

# Loop to find the required sample size
while(power < threshold && df < 200){

  # Calculate the critical chi-square value for the given alpha
  x <- qchisq(alpha, df = df)

  xscaled <- x * sigma0_sq / sigma_a_sq
  power <- pchisq(xscaled, df = df)

  if(power < threshold){
    n <- n + 1
    df <- df + 1
```

```
  }
}

# Output the required sample size
if(power >= threshold){
  cat("Required sample size (n):", n, "\n")
} else {
  cat("Sample size did not reach required power within n=100.\n")
}

## Required sample size (n): 37

x = qchisq(.01, 36)
print(x)

## [1] 19.23268

xadj <- x*225 / 100
print(xadj)

## [1] 43.27352

power <- pchisq(xadj, df=36, lower.tail=TRUE)

print(power)

## [1] 0.8113461
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

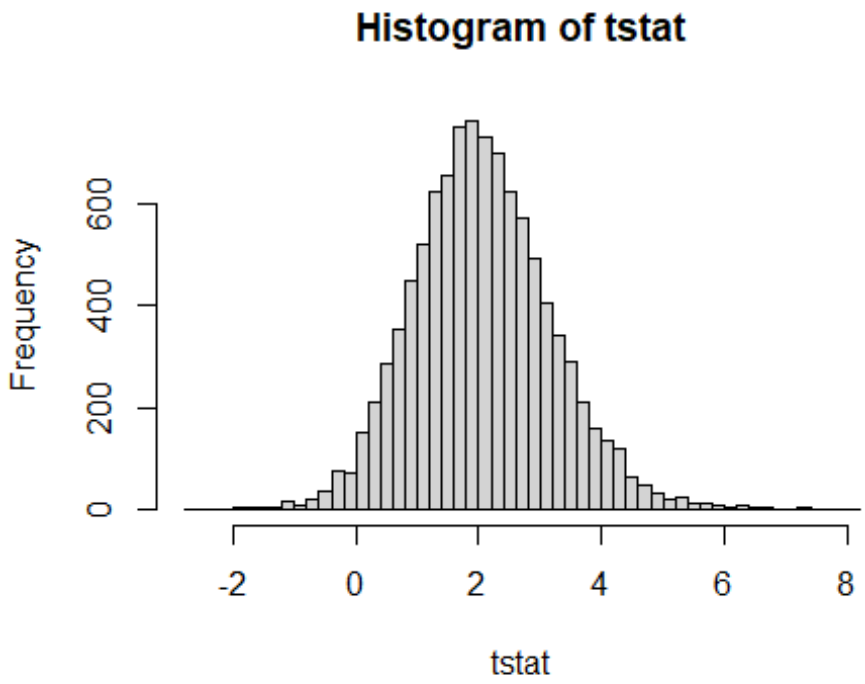
#4a:

```
1-pt(5.316, 19)

## [1] 1.972177e-05
```

#4b

```
n = 20
sd = 45
true_mean <- 1180

nloop = 10000

tstat <- numeric(nloop)

for(iloop in 1:nloop){
  y=rnorm(n, true_mean, sd)
  samp_mean = mean(y)
  samp_sd = sd(y)
  tstat[iloop] = (samp_mean - 1160)/(samp_sd /sqrt(n))
```

```
}

hist(tstat, br=50)
```




```
sum(tstat > qt(1-.05,n-1))/nloop

## [1] 0.6042
```

#5b

```
2*(1-pt(.978, df=24))

## [1] 0.3378304
```

#5c

```
2* (1-pf(2.1199, df1=11, df2=13))

## [1] 0.1987924
```