Homework Assignment 3

Elham

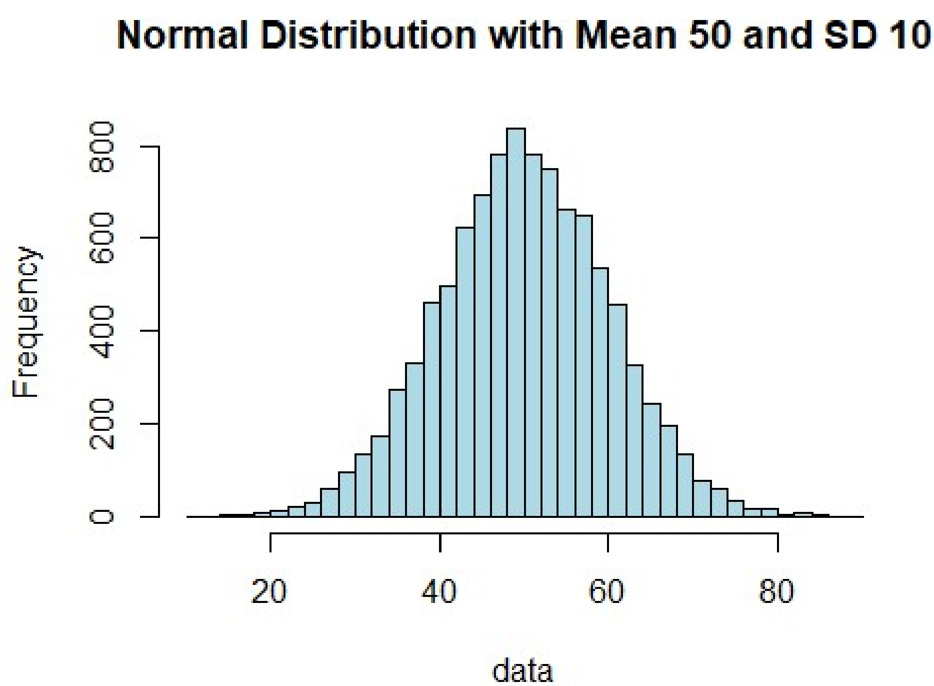
2023-02-26

1. Use the following commands:

set.seed(123) data <- rnorm(10000, mean = 50, sd = 10)

The commands generate a normal distribution with a mean of 50 and a standard deviation of 10. Create a histogram to show the range of values that covers the middle 95% of this distribution, using the title of “Normal Distribution with Mean 50 and SD 10”, light blue color, and breaks equal to 30.

```
set.seed(123)
data <- rnorm(10000, mean = 50, sd = 10)
hist(data, breaks = 30, col = "lightblue", main = "Normal Distribution with
Mean 50 and SD 10")
```



2. Download the data sets sampledata.csv with 100 numeric variables, name it mydata and save it to whatever directory you are using for this question. (a) Use a for loop to calculate the mean and standard deviation for each variable in mydata. (b) Use an if statement to identify the variables with a mean greater than 10.5 and store their names in a vector.

```r
mydata<-read.csv("C:\\Users\\elham\\OneDrive\\Desktop\\sampledata.csv")

# Use a for loop to calculate the mean and standard deviation for each
variable
for (i in 1:ncol(mydata)) {
  means <- rep(NA, ncol(mydata))
  sds <- rep(NA, ncol(mydata))
}

#means <- rep(NA, ncol(mydata)) This method is better but students may not
use the rep() function
#sds <- rep(NA, ncol(mydata)) This method is better but students may not use
the rep() function

for (i in 1:ncol(mydata)) {
  means[i] <- mean(mydata[[i]])
  sds[i] <- sd(mydata[[i]])
}

# Use an if statement to identify the variables with a mean greater than 10.5
and store their names in a vector
high_mean_vars <- c()
for (i in 1:length(means)) {

  means[i]<- mean(mydata[,i])
  if (means[i] > 10.5) {
    high_mean_vars <- c(high_mean_vars, names(mydata[i]))
  }
}

high_mean_vars

## [1] 25.5
## [1] 25.50000 10.69294
## [1] 25.50000 10.69294 10.96471
## [1] 25.50000 10.69294 10.96471 10.70114
## [1] 25.50000 10.69294 10.96471 10.70114 10.55554
## [1] 25.50000 10.69294 10.96471 10.70114 10.55554 10.66256
```

3. Use the dataset mydata in question 2.

(a) Calculate the mean for each variable.

(b) Calculate the 90% confidence interval for each variable

(c) Print the results using a matrix with three columns named ("Mean", "Lower_CI", and "Upper_CI"). Hint: You need to use mydata and create a matrix with three columns. The number of rows equals the number of rows in mydata. Next, you need to apply for loops to calculate the mean and the 90% confidence interval. Finally, fill the matrix with the results.

```r
# Generate sample data
mydata<-read.csv("C:\\Users\\elham\\OneDrive\\Desktop\\sampledata.csv")
```

```r
# Calculate the mean and the 90% confidence interval for each variable
result <- matrix(NA, nrow = 100, ncol = 3) # or fill matrix with zero ===>
result <- matrix(0, nrow = 100, ncol = 3)
colnames(result) <- c("Mean", "Lower_CI", "Upper_CI")
for (i in 1:100) {
  n <- length(mydata[,i])
  mean <- mean(mydata[,i])
  std_error <- sd(mydata[,i])/sqrt(n)
  margin_of_error <- qnorm(0.95) * std_error
  ci <- c(mean - margin_of_error, mean + margin_of_error)
  result[i,] <- c(mean, ci)
}

# Print the results
print(result)

##             Mean  Lower_CI  Upper_CI
##   [1,] 25.500000 22.109047 28.890953
##   [2,] 10.692941 10.181693 11.204189
##   [3,]  9.925052  9.411725 10.438378
##   [4,] 10.315030  9.904809 10.725252
##   [5,] 10.964712 10.506939 11.422485
##   [6,] 10.233889  9.769067 10.698710
##   [7,]  9.870636  9.386218 10.355054
##   [8,]  9.805371  9.336834 10.273908
##   [9,]  9.642414  9.231887 10.052941
##  [10,] 10.022282  9.470403 10.574161
##  [11,] 10.213038  9.804171 10.621906
##  [12,] 10.307688  9.872073 10.743303
##  [13,]  9.814616  9.340416 10.288815
##  [14,]  9.893660  9.414566 10.372753
##  [15,]  9.749170  9.202943 10.295396
##  [16,] 10.096851  9.575277 10.618425
##  [17,]  9.893779  9.400550 10.387009
##  [18,] 10.701142 10.231619 11.170665
##  [19,] 10.261952  9.799486 10.724418
##  [20,] 10.075299  9.657817 10.492780
##  [21,] 10.049098  9.640617 10.457579
##  [22,]  9.717226  9.315213 10.119239
##  [23,] 10.292880  9.831213 10.754546
##  [24,] 10.133018  9.667486 10.598550
##  [25,]  9.910439  9.456830 10.364049
##  [26,]  9.739426  9.280694 10.198158
##  [27,] 10.199722  9.818802 10.580642
##  [28,] 10.555536 10.065362 11.045710
##  [29,] 10.363437  9.918248 10.808626
##  [30,] 10.206064  9.756908 10.655220
##  [31,]  9.816106  9.392120 10.240092
##  [32,] 10.662563 10.225692 11.099433
```

```
##  [33,] 10.334365  9.881280 10.787450
##  [34,]  9.612551  9.102640 10.122462
##  [35,] 10.091998  9.673691 10.510306
##  [36,] 10.139857  9.587420 10.692294
##  [37,] 10.093518  9.600676 10.586360
##  [38,]  9.607415  9.095977 10.118853
##  [39,] 10.003706  9.569058 10.438354
##  [40,] 10.056218  9.523535 10.588901
##  [41,]  9.846925  9.397945 10.295905
##  [42,]  9.909246  9.421506 10.396987
##  [43,]  9.750470  9.246419 10.254522
##  [44,]  9.754773  9.314449 10.195098
##  [45,] 10.077017  9.608236 10.545799
##  [46,] 10.050344  9.707846 10.392842
##  [47,] 10.188217  9.665742 10.710692
##  [48,]  9.802783  9.330258 10.275308
##  [49,]  9.552618  9.085776 10.019460
##  [50,]  9.917454  9.418617 10.416291
##  [51,] 10.169843  9.712943 10.626743
##  [52,]  9.856027  9.442178 10.269877
##  [53,]  9.299442  8.817962  9.780922
##  [54,] 10.045699  9.609676 10.481721
##  [55,] 10.184347  9.710665 10.658028
##  [56,]  9.885970  9.486444 10.285496
##  [57,]  9.852738  9.343376 10.362101
##  [58,]  9.838533  9.357855 10.319212
##  [59,]  9.832220  9.317569 10.346871
##  [60,]  9.674892  9.266072 10.083712
##  [61,]  9.911734  9.446591 10.376878
##  [62,]  9.669543  9.214837 10.124249
##  [63,]  9.744509  9.314965 10.174053
##  [64,] 10.245969  9.774697 10.717241
##  [65,]  9.934396  9.539301 10.329490
##  [66,] 10.177343  9.766709 10.587978
##  [67,] 10.058952  9.527365 10.590538
##  [68,]  9.857526  9.345167 10.369885
##  [69,]  9.857805  9.372002 10.343609
##  [70,]  9.600571  9.177927 10.023214
##  [71,]  9.696385  9.236924 10.155845
##  [72,] 10.233076  9.774576 10.691577
##  [73,] 10.027390  9.511637 10.543143
##  [74,]  9.812836  9.325714 10.299958
##  [75,] 10.191553  9.690988 10.692118
##  [76,] 10.054493  9.637667 10.471319
##  [77,] 10.067387  9.589966 10.544808
##  [78,]  9.817793  9.363362 10.272225
##  [79,]  9.589308  9.090522 10.088093
##  [80,] 10.188418  9.775623 10.601214
##  [81,] 10.006802  9.527635 10.485968
##  [82,] 10.301417  9.794910 10.807925
```

```
##  [83,]  9.953892  9.398281 10.509504
##  [84,]  9.708866  9.209922 10.207811
##  [85,] 10.345541  9.858580 10.832503
##  [86,] 10.195723  9.747876 10.643570
##  [87,]  9.921911  9.451854 10.391969
##  [88,]  9.897960  9.456836 10.339083
##  [89,]  9.791028  9.308563 10.273494
##  [90,] 10.344184  9.953218 10.735149
##  [91,]  9.690279  9.219848 10.160710
##  [92,] 10.294662  9.858699 10.730626
##  [93,] 10.292739  9.906363 10.679114
##  [94,] 10.163757  9.689257 10.638257
##  [95,] 10.339230  9.819377 10.859083
##  [96,] 10.244536  9.826830 10.662243
##  [97,]  9.934854  9.438846 10.430863
##  [98,] 10.167383  9.758544 10.576222
##  [99,] 10.041437  9.548150 10.534724
## [100,]  9.438359  8.949039  9.927678
```

Practice Question (Review of concepts throughout the entire course): Consider the built-in data set UCBAdmissions.

1. If we are interested in the proportion of people that apply to Berkeley University and get accepted, what is the population of interest and what is the parameter of interest?

Population	Variable of Interest	Parameter
People who apply to Berely University	Whether or not an individual is accepted	Proportion of population who get accepted

2. Using the command ?UCBAdmissions, determine the variables in the dataset and describe what kind of variables they are.

Three variables. All categorical.

1. Admit: Admitted, Rejected
2.Geneder: Male, Female
3. Dept: A, B, C, D, E, F

3. Create a variable in R called *totalAdmissions* which contains the total number of students who were admitted to the university (across all genders and departments).

totalAddmisions = sum(UCBAdmissions[1, ,])

1755

4. Create a variable in R called *totalRejections* which contains the total number of students who were rejected to the university (across all genders and departments).

totalRejections = sum(UCBAdmissions[2, ,])
or
totalRejections = sum(UCBAdmissions["Rejected", ,])
2771

5. Create a variable in R called *totalApplicants* which contains the total number of students who applied to the university in our sample.

totalApplicants= totalAddmisions+ totalRejections
4526

6. What is the observed value of the statistic we should use to estimate the population parameter of interest?

phat= totalAddmisions/totalApplicants

0.3877596

7. What is the estimated standard error for $\hat{p}$?

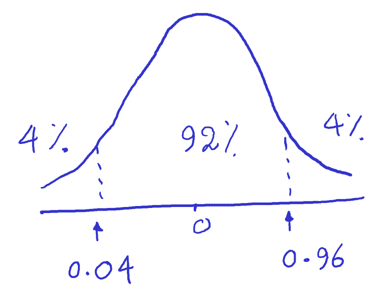
ese = sqrt(phat*(1-phat)/totalApplicants) $\rightarrow$ estimated standard error = $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

0.007242442

8. What is the critical value for a 92% confidence interval for p?

cv= qnorm(0.96) $\rightarrow$ Critical value $\begin{cases} -\text{qnorm}(0.04) \\ \text{or} \\ \text{qnorm}(0.96) \end{cases}$

1.750686



9. What is the margin of error for our estimate?

moe= cv*ese

0.01254007

10. Compare that to result of the approximate margin of error formula we learned earlier in the course.

Approximate moe feom previous formula in chapter 3: $\frac{1}{\sqrt{n}} \simeq \frac{1}{\sqrt{4526}} = 0.01486424$

Note: Don't use $\frac{1}{\sqrt{n}}$ for moe unless you are being asked.

11. Determine a 92% confidence interval for the true value of the population proportion.

Upper_bound= phat+moe

0.4004389

Lower_Bound= phat-moe

0.3750804

Note: It seems that students often get confused about how to use the quantile() and qnorm() functions in R.

The answer depends on the type of question being asked. For example, if the question asks for a critical value for a sample proportion or sample mean, then the qnorm() function should be used. It is not recommended to use the quantile() function to calculate critical values for hypothesis testing. Instead, you should use the qnorm() function, or other similar functions such as qt() for the t-distribution or qchisq() for the chi-squared distribution.

The quantile() function is typically used to compute quantiles of a given dataset, but not for critical values in hypothesis testing. The qnorm() function can be used to calculate quantiles for the standard normal distribution.

Therefore, whether to use qnorm() or quantile() depends on whether the question asks for a critical value for any statistic or quantiles for the standard normal distribution. If you are not sure whether the given dataset has a normal distribution and the question asks for quantiles of a given dataset, then you should use the quantile() function.