Project 3

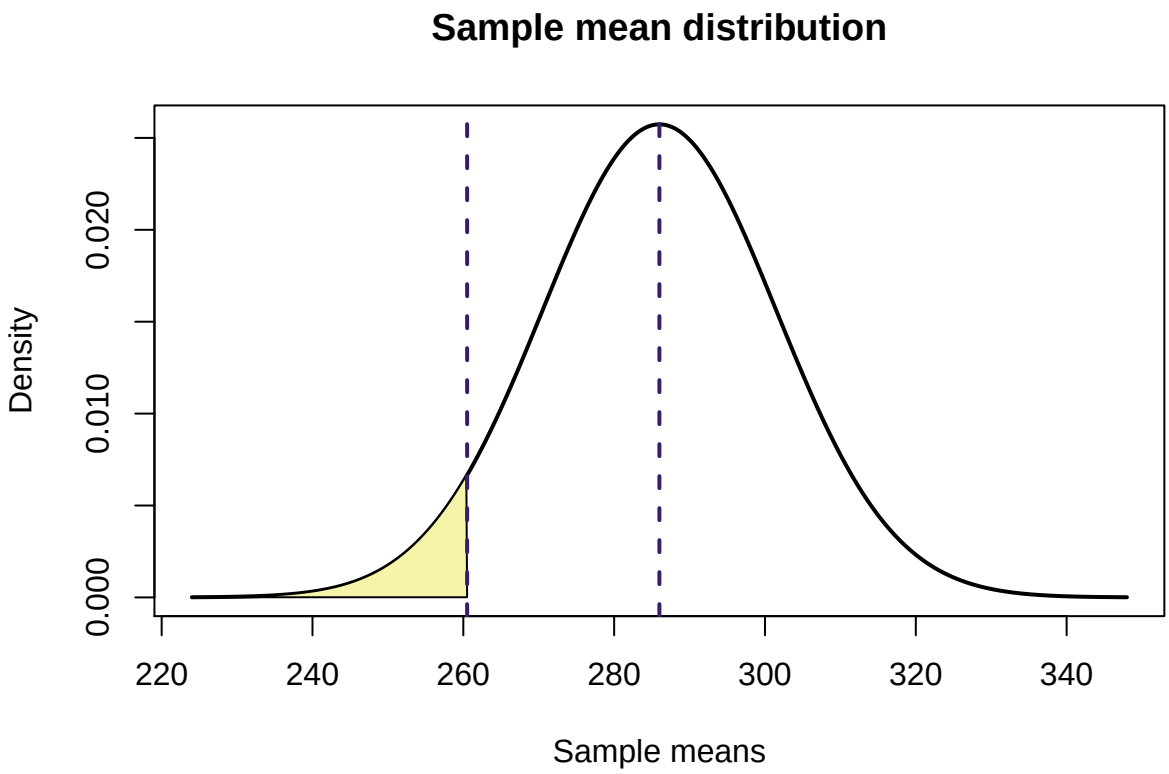
Wenjing (Audrey) Yang, Youhyun (Lisa) Kang, Yang Wang, Lavanya Makdani

11/17/2020

Problem 3.1

(i)

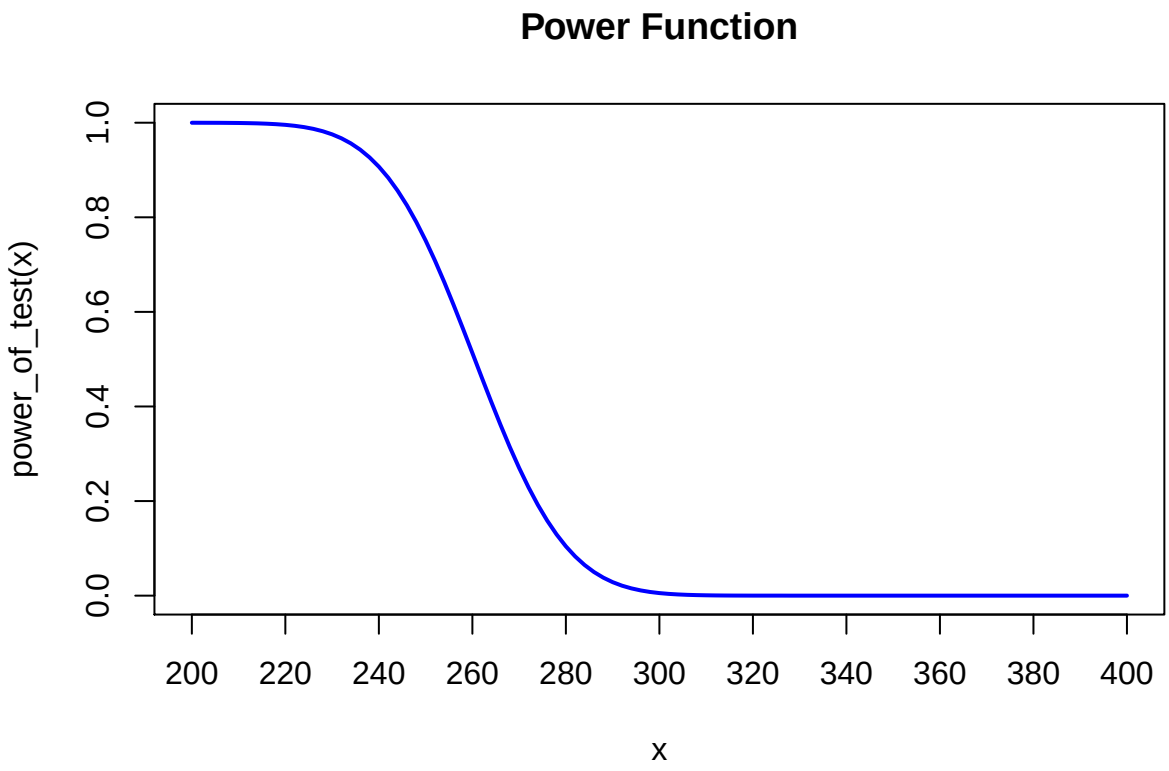
```
#distribution of sample averages
n <- 100
mu <- 286
s <- 155/10
x <-seq(mu - 4 *s , mu + 4 *s, 0.1)
y = dnorm(x,mean = mu,sd = s)

#plot
plot(x,
     y,
     type = "l",
     main = "Sample mean distribution",
     xlab = "Sample means",
     ylab = "Density",
     lwd = 2)

#find upper bound
upper_bound <- 286 + qnorm(0.05)*s

#Shade the region below the normal density function
#to the left of the upper bound
polygon(c(min(x), x[x <= upper_bound]),
        c(y[x <= upper_bound],
           y[x == min(x)]), col = '#f5f4a9' )


#draw a vertical line of population mean
abline(v = 286, lty = 2, lwd = 2, col= "#381e69")


#draw a vertical line of upper bound of the rejection region
abline(v = upper_bound, lty = 2, lwd = 2, col= "#381e69")
```



(ii)

```r
# Find upper bound and rejection region
n <- 100
x <- rnorm(n,mean=286,sd=155)
x_bar <- mean(x)
s <- 155/10
z <- (x_bar-286)/s
upper_bound <- 286 + qnorm(0.05)*s

# Write power of test function of the alternative population mean
power_of_test <- function(alter_mean){
  z_new <- (upper_bound - alter_mean)/s
  return(pnorm(z_new))
}

# Draw the graph of the power function
std <- 155
n <- 100
mean <- 286
alpha <- 0.05
upper_bound <- mean + qnorm(alpha)*std/sqrt(n)

curve(power_of_test(x),
```

```
      from = 200,
      to = 400,
      col = "blue",
      main = "Power Function",
      xaxt = "n",
      lwd = 2)

axis(1, at = seq(200, 400, by = 20))
```

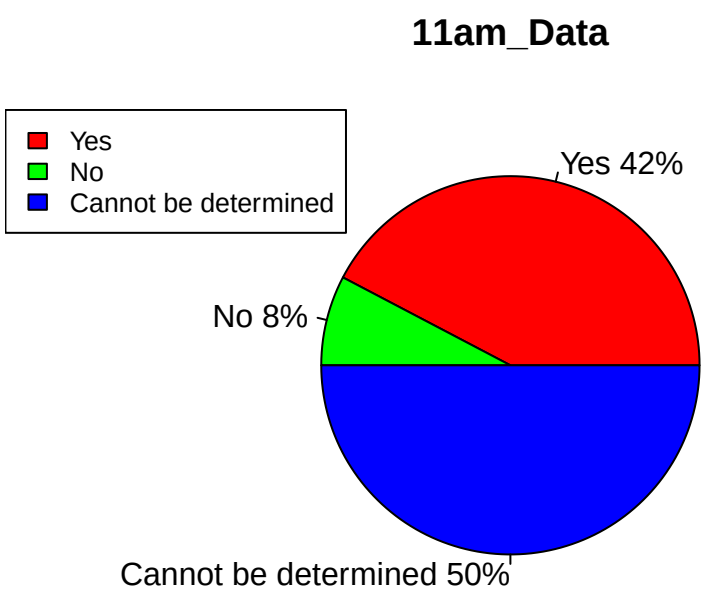


Problem 3.2

(i)

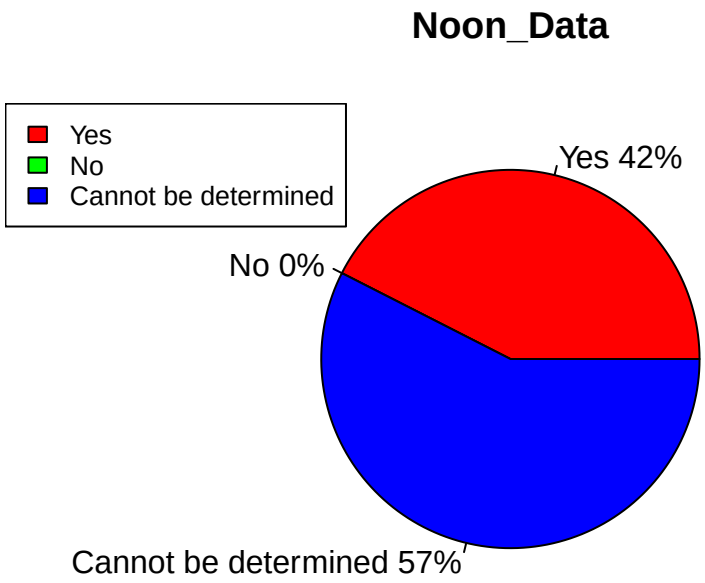
```
# extracting data from the given csv file for 11am section
# created two different files for 11am and 12pm from given csv file
am_data = read.csv("logic-puzzle-abridged-11.csv", header = TRUE)
am_yes = sum(am_data$'Answer' == "Yes")
am_no = sum(am_data$'Answer' == "No")
am_cannot = sum(am_data$'Answer' == "Cannot be determined")
am_total = sum(am_data$'Answer' != "")

#calculating p_am
```

```
p_am = am_yes / am_total

#data for the graph
x = c(am_yes, am_no, am_cannot)
labels_x = c("Yes", "No", "Cannot be determined")
pct_x = round(x/sum(x)*100)           #calculates the percentages and
                                      #stores in in variable pct_x
lbls_x = paste(labels_x, pct_x)       #add percents to labels_x
lbls_x = paste(lbls_x, "%", sep = "") #add % to labels

#pie Chart for 11am section
pie(x, labels = lbls_x, col = rainbow(length(lbls_x)), main = "11am_Data")
legend("topleft", c("Yes", "No", "Cannot be determined"), cex=0.8, fill= rainbow(length(x)))
```




```
# extracting data from the given csv file for 12pm section
pm_data = read.csv("logic-puzzle-abridged-12.csv", header = TRUE)
pm_yes = sum(pm_data$'Answer' == "Yes")
pm_no = sum(pm_data$'Answer' == "No")
pm_cannot = sum(pm_data$'Answer' == "Cannot be determined")
pm_total = sum(pm_data$'Answer' != "")

#calculating p_pm
p_pm = pm_yes / pm_total

# data for the graph
```

```r
y = c(pm_yes, pm_no, pm_cannot)
labels_y = c("Yes", "No", "Cannot be determined")
pct_y = round(y/sum(y)*100)           #calculates the percentages and stores
                                      #in in variable pct_y
lbls_y = paste(labels_y, pct_y)       #add percents to labels
lbls_y = paste(lbls_y, "%", sep = "") #add % to labels

#pie Chart for 12pm (Noon) section
pie(y, labels = lbls_y, col = rainbow(length(lbls_y)), main = "Noon_Data")
legend("topleft", c("Yes", "No", "Cannot be determined"), cex=0.8, fill= rainbow(length(y)))
```




```r
#table for showing total data
table = matrix(c(am_yes,
                 am_no,
                 am_cannot,
                 am_total,
                 pm_yes,
                 pm_no,
                 pm_cannot,
                 pm_total,
                 (am_yes + pm_yes),
                 (am_no + pm_no),
                 (am_cannot + pm_cannot),
                 (am_total+pm_total)),
               ncol=4,
```

```
               nrow = 3,
               byrow = TRUE)
colnames(table) = c("Yes", "No", "CBD", "Total")
rownames(table) = c("11am", "12pm","Total")
table = as.table(table)
table
```

```
##       Yes No CBD Total
## 11am   22  4  26    52
## 12pm   17  0  23    40
## Total  39  4  49    92
```

```
#Hypothesis testing
test_stats <- p_am - p_pm
p_hat <- ((am_yes+pm_yes)/(am_total+pm_total))
z_stats <- (test_stats / sqrt(p_hat * (1 - p_hat)*((1/am_total)+(1/pm_total))))
p_value <- 2*pnorm(z_stats)   #calculating p value for two proportions
print(p_value)
```

```
## [1] 0.9852372
```

(ii)

```
#setwd("/Users/nitin/Desktop")
#answers = read.csv(file = "logic-puzzle.csv", sep= ",")
#yes_people = subset(answers, What.s.the.answer.to.the.above.logic.puzzle. == "Yes")
#used the above functions to check how many people responded "Yes"
yes_people = 39
total_people = 92
no_people = 4
unsure_people = 49
percentage = yes_people / total_people
a = (percentage - 0.2)
b = sqrt((0.2 * (1-0.2)) / total_people)
z = a / b
p_value =  (1 - pnorm(z))
print(p_value)
```

```
## [1] 3.953322e-08
```

Problem 3.3

(i)

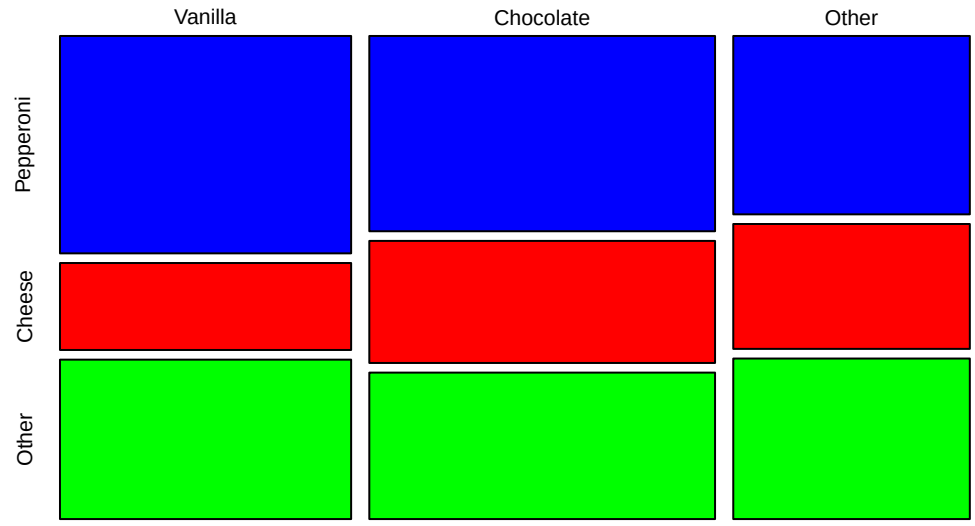
```
#pizza and ice cream data extracted from given csv file
pi_tw = matrix(c(15,6,11,32,16,10,12,38,10,7,9,26,41,23,32,96),
               ncol=4,
```

```
               nrow = 4,
               byrow = TRUE)   #data for two way table
colnames(pi_tw) = c("Pepperoni", "Cheese", "Other","Total")
rownames(pi_tw) = c("Vanilla", "Chocolate","Other","Total")
pi_tw = as.table(pi_tw)
pi_tw
```

```
##           Pepperoni Cheese Other Total
## Vanilla          15      6    11    32
## Chocolate        16     10    12    38
## Other            10      7     9    26
## Total            41     23    32    96
```

(ii)

```
pi_tw = matrix(c(15,6,11,16,10,12,10,7,9), ncol=3, nrow = 3, byrow = TRUE)
colnames(pi_tw) = c("Pepperoni", "Cheese", "Other")
rownames(pi_tw) = c("Vanilla", "Chocolate","Other")
pi_tw = as.table(pi_tw)
plot(pi_tw, main = "Pizza and Ice Cream", color=c("blue", "red", "green"), shade = FALSE)
```



(iii)

```
pi_tw = matrix(c(15,6,11,16,10,12,10,7,9), ncol=3, nrow = 3, byrow = TRUE)
colnames(pi_tw) = c("Pepperoni", "Cheese", "Other")
rownames(pi_tw) = c("Vanilla", "Chocolate","Other")
pi_tw = as.table(pi_tw)

test <- chisq.test(pi_tw)
test$observed
```

```
##           Pepperoni Cheese Other
## Vanilla          15      6    11
## Chocolate        16     10    12
## Other            10      7     9
```

```
test$expected
```

```
##           Pepperoni   Cheese     Other
## Vanilla    13.66667 7.666667 10.666667
## Chocolate  16.22917 9.104167 12.666667
## Other      11.10417 6.229167  8.666667
```

```
test
```

```
##
##  Pearson's Chi-squared test
##
## data:  pi_tw
## X-squared = 0.84729, df = 4, p-value = 0.932
```