Lab6_stat123

Koki Itagaki

2023-03-01

#1. Generate a dataset of student grades, with columns for the student's name, their midterm score, and their final exam score as below. #name = c("Alice", "Bob", "Charlie", "David", "Emily", "Frank", "Grace", #"Henry", "Isabel", "John","Karen", "Liam", "Megan", "Nate", "Olivia"), #midterm = c(80, 70, 91, 85, 55, 80, 95, 50, 65, 75, 80, 85, 90, 75, 80), #final = c(85, 80, 70, 90, 55, 85, 90, 75, 85, 70, 80, 80, 70, 60, 75)

```
#(a) Take this dataset as input and returns a vector of the final grades for
#each student. (weighted avetrage of the midterm is(0.4)and final score is
0.6)
data<-data.frame(name = c("Alice", "Bob", "Charlie", "David", "Emily",
"Frank", "Grace","Henry", "Isabel", "John","Karen", "Liam", "Megan", "Nate",
"Olivia"),
midterm = c(80, 70, 91, 85, 55, 80, 95, 50, 65, 75, 80, 85, 90, 75, 80),
final = c(85, 80, 70, 90, 55, 85, 90, 75, 85, 70, 80, 80, 70, 60, 75))
data
```

```
##       name midterm final
## 1    Alice      80    85
## 2      Bob      70    80
## 3  Charlie      91    70
## 4    David      85    90
## 5    Emily      55    55
## 6    Frank      80    85
## 7    Grace      95    90
## 8    Henry      50    75
## 9   Isabel      65    85
## 10    John      75    70
## 11   Karen      80    80
## 12    Liam      85    80
## 13   Megan      90    70
## 14    Nate      75    60
## 15  Olivia      80    75
```

```
#To get final grades
grades<-data$midterm*0.4 + 0.6*data$final

#(b) Use if statements to assign a letter grade (A, B, C, D, or F) to the
#student named "Charlie. " (weighted average of the midterm is (o.4) and
#final scores is (0.6)), with the following criteria:
#A: grade >= 90
#B: grade >= 80 and grade < 90
#C: grade >= 70 and grade < 80
```

```r
#D: grade >= 60 and grade < 70
#F: grade < 60

letter_grades <- character(1)
letter_grades
```

```
## [1] ""
```

```r
if(grades[3]>= 90){

}else if(grades[3]>= 90){
  letter_grades<- "A"
}else if(grades[3]>= 80){
  letter_grades<- "B"
}else if(grades[3]>= 70){
  letter_grades<- "C"
}else if(grades[3]>= 60){
  letter_grades<- "D"
}else{
  letter_grades<- "F"

}
print(letter_grades)
```

```
## [1] "C"
```

```r
#Bonus
#create vecttor first
letter_grade<-character(length(grades))
letter_grade<-ifelse(grades >= 90, "A",
              ifelse(grades >= 80, "B",
              ifelse(grades >= 70, "C",
              ifelse(grades >= 60, "D","F"))))
letter_grade
```

```
##  [1] "B" "C" "C" "B" "F" "B" "A" "D" "C" "C" "B" "B" "C" "D" "C"
```

```r
#Add new column
data$grade <-letter_grade
data
```

```
##       name midterm final grade
## 1    Alice      80    85     B
## 2      Bob      70    80     C
## 3  Charlie      91    70     C
## 4    David      85    90     B
## 5    Emily      55    55     F
## 6    Frank      80    85     B
## 7    Grace      95    90     A
## 8    Henry      50    75     D
## 9   Isabel      65    85     C
```

```
## 10    John      75    70     C
## 11   Karen      80    80     B
## 12    Liam      85    80     B
## 13   Megan      90    70     C
## 14    Nate      75    60     D
## 15  Olivia      80    75     C
```

#2. You have a variable x that contains a numeric value. You can consider any #numeric value for x.

```
#(a) Check whether x is positive, negative, or zero.
x<- -9
if(x>0){
  print("x is positive")
}else if(x<0){
  print("X is negative")
}else{
  print("X is zero")
}

## [1] "X is negative"

#(b) Considering the numeric value of x, print out x “is positive” or x “is
#negative” or x “is zero.”
```

#3. Load the normal_distribution.csv dataset and save it as nd.

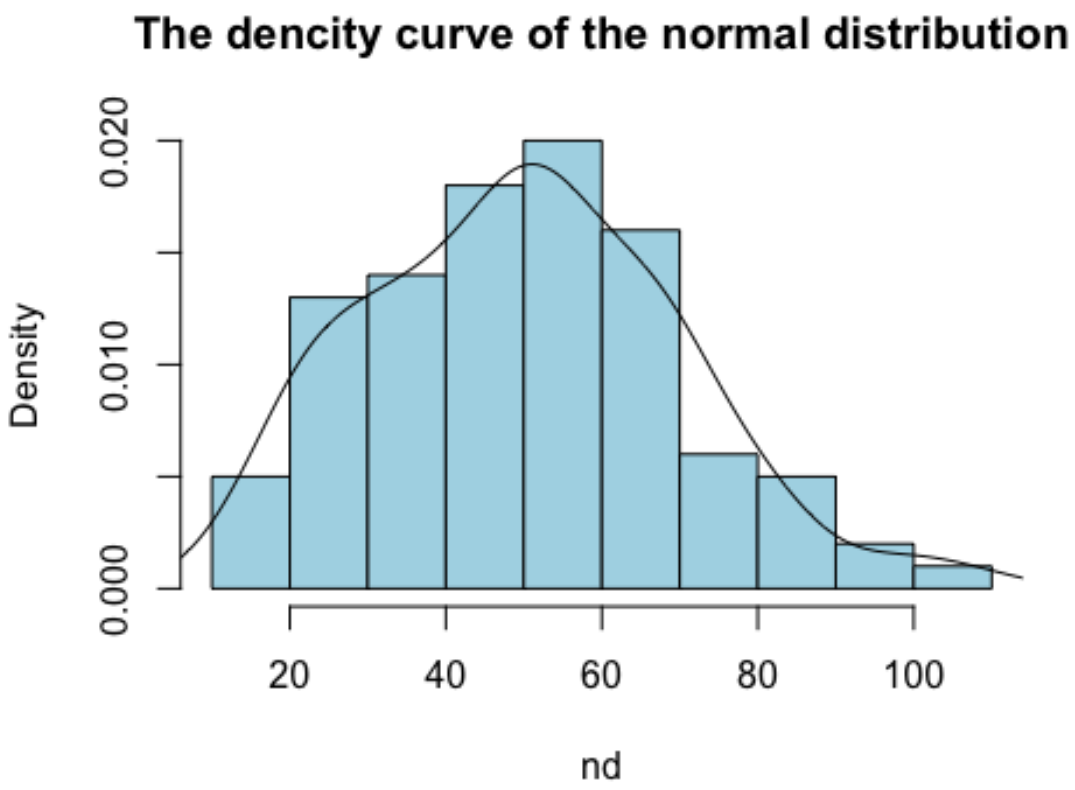
```
nd<-read.csv("/Users/itagakikouki/stat123/lab6/normal_distribution.csv")
dim(nd)

## [1] 100   1

head(nd)

##     normal
## 1 51.09344
## 2 56.73150
## 3 28.81101
## 4 52.94184
## 5 28.75654
## 6 54.35247

nd<-as.numeric(nd$normal)
#(a) Generate the density curve of the nd using the density() function.
hist(nd, main = "The dencity curve of the normal distribution", prob = TRUE,
     col = "lightblue")
#Density function to plot the dencity of the curve
lines(density(nd))
```




```
#(b) Find the mean of nd and save it as mu.
mu<- mean(nd)
mu

## [1] 49.91465

#(c) find the standard deviation of nd and save it as sig.
sig<-sd(nd)
sig

## [1] 19.86311

#(d) Use the function quantile() and the values mu and sig to find the value
#in nd that is greater than 65% of the values in nd.
quantile(nd, .65)

##      65%
## 56.60207
```