Worksheet1

stat414

2024-09-14

```r
#===================== QUESTION #1: =====================#
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.3.3
```

```r
#library(Rcmdr)

data <- read_excel("Profit.xlsx") #import excel data

dataframe <- data.frame(data) #convert excel sheet to dataframe
dataframe
```

```
##              REGION PROFIT POPULATION STORES  AREA BONUS
## 1         Andromeda   1011      3.881    213 16.96     1
## 2            Antlia   1318      3.141    158  7.31     1
## 3            Aquila   1556      3.766    203  7.81     1
## 4               Ara   1521      4.587    170  7.31     1
## 5            Auriga    979      3.648    142 19.84     1
## 6            Bootes   1290      3.456    159 12.37     1
## 7            Caelum   1596      3.695    178  6.15     1
## 8    Camelopardalis   1155      3.609    182 14.21     1
## 9            Carina   1412      3.801    181  7.45     1
## 10       Cassiopeia   1194      3.322    148 14.43     1
## 11        Centaurus   1054      5.124    227  6.12     0
## 12          Cepheus   1157      4.158    139 11.71     1
## 13            Cetus   1001      3.887    179  9.36     0
## 14         Circinus    831      2.230    124 19.14     1
## 15           Corvus    857      4.468    205 11.75     0
## 16             Crux    188      0.297     85 40.34     1
## 17           Cygnus   1030      4.224    211  7.16     0
## 18        Delphinus   1331      3.427    145  9.37     1
## 19           Dorado    643      4.310    205  7.62     1
## 20            Draco    992      2.370    166 27.54     1
## 21         Equuleus    795      3.903    149 15.97     1
## 22         Eridanus   1340      3.423    186 12.97     1
## 23           Fornax    689      2.390    141 17.36     0
## 24             Grus   1726      4.947    233  6.24     1
## 25         Hercules   1056      4.166    176 11.20     0
## 26       Horologium    989      4.063    187 18.09     1
## 27            Hydra    895      3.105    131 13.32     1
## 28          Lacerta   1028      4.116    170 14.97     0
```

```
## 29          Lynx    771       1.510    144 21.92     1
## 30          Lyra    484       0.741    126 34.91     1
## 31   Microscopium    917       5.260    234  8.46     0
## 32     Monoceros   1786       5.744    210  7.52     0
## 33         Musca   1063       2.703    141 14.43     1
## 34         Norma   1001       3.583    158 15.37     0
## 35        Octans   1052       4.469    167 11.20     0
## 36     Ophiuchus   1610       4.951    174  7.20     1
## 37         Orion   1486       3.474    211 13.49     1
## 38          Pavo   1576       4.637    172  6.56     1
## 39       Pegasus   1665       3.900    185  9.35     1
## 40       Perseus    878       3.766    166 11.12     0
## 41       Phoenix    849       3.876    189 10.58     0
## 42        Puppis    775       3.753    164 17.82     0
## 43         Pyxis   1012       4.449    193 10.03     0
## 44       Sagitta   1436       4.680    157 10.01     1
## 45       Serpens    798       4.806    200 10.70     0
## 46   Telescopium    519       2.367    142 24.38     0
## 47    Triangulum   1701       5.563    199  6.57     0
## 48        Tucana   1387       4.357    166  6.64     1
## 49          Vela   1717       4.670    221  9.24     1
## 50        Volans   1032       3.993    180 11.62     0
## 51     Vulpecula    973       3.923    193 12.85     0
```

```r
profits <- as.numeric(dataframe[,2]) #extract column 2 as numeric values
max(profits)
```

```
## [1] 1786
```

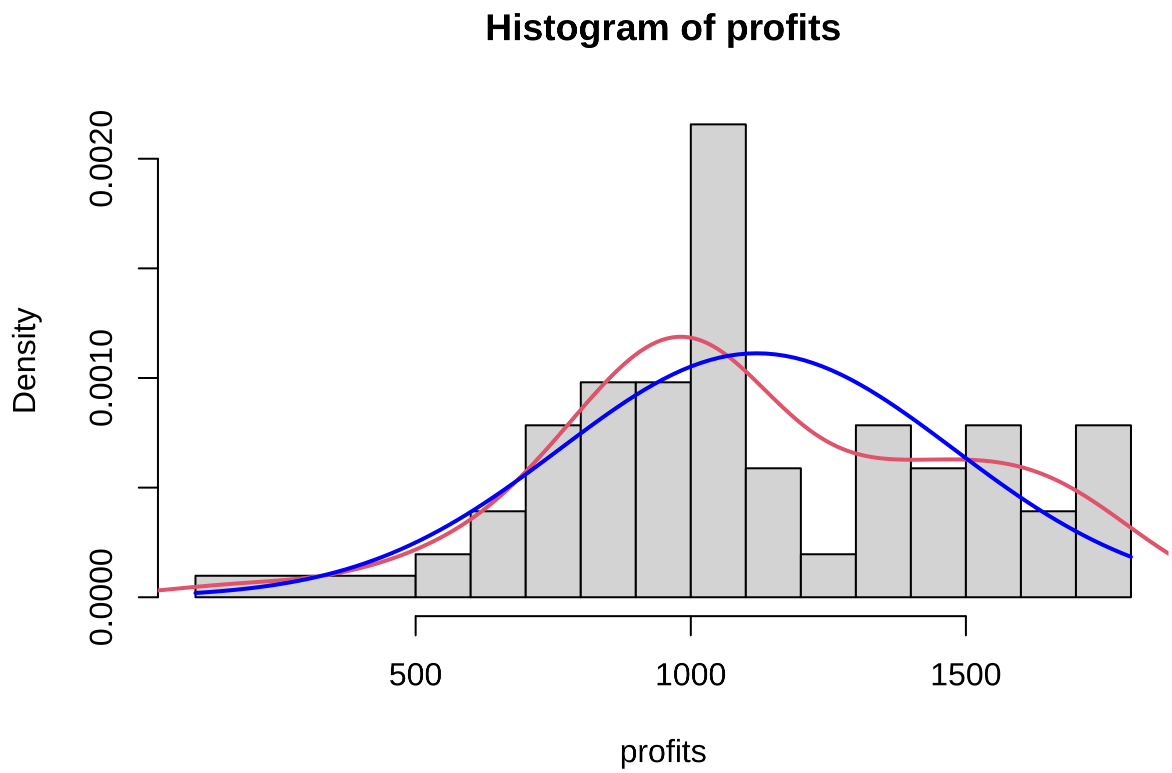
```r
min(profits)
```

```
## [1] 188
```

```r
profits
```

```
##  [1] 1011 1318 1556 1521  979 1290 1596 1155 1412 1194 1054 1157 1001  831  857
## [16]  188 1030 1331  643  992  795 1340  689 1726 1056  989  895 1028  771  484
## [31]  917 1786 1063 1001 1052 1610 1486 1576 1665  878  849  775 1012 1436  798
## [46]  519 1701 1387 1717 1032  973
```

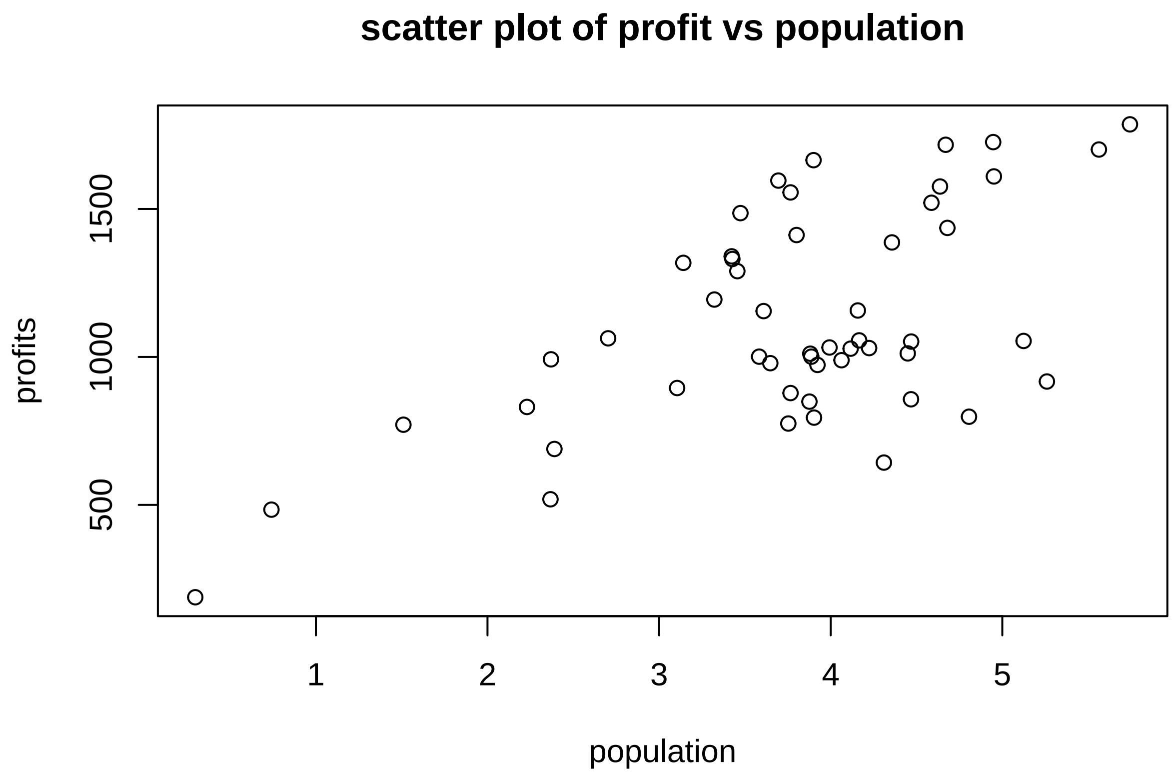
```r
#Histogram of column 2 profits
hist(profits,probability=TRUE,breaks= c(100,500,600,700,800,900,1000,
                       1100,1200,1300,1400,1500,1600,1700,1800))

#Superimpose normal line onto histogram
lines(density(profits), col = 2, lwd = 2)
curve(dnorm(x, mean=mean(profits), sd=sd(profits)), lwd=2, col="blue", add=TRUE)
```




```r
#Plot population vs profits in a scatterplot
population <- as.numeric(dataframe[,3]) #extract column 3 as numeric
population
```

```
##  [1] 3.881 3.141 3.766 4.587 3.648 3.456 3.695 3.609 3.801 3.322 5.124 4.158
## [13] 3.887 2.230 4.468 0.297 4.224 3.427 4.310 2.370 3.903 3.423 2.390 4.947
## [25] 4.166 4.063 3.105 4.116 1.510 0.741 5.260 5.744 2.703 3.583 4.469 4.951
## [37] 3.474 4.637 3.900 3.766 3.876 3.753 4.449 4.680 4.806 2.367 5.563 4.357
## [49] 4.670 3.993 3.923
```

```r
plot(population, profits, main="scatter plot of profit vs population")
```




```r
#Summary of data
SUM <- summary(dataframe)
SUM
```

```
##     REGION              PROFIT         POPULATION         STORES
##  Length:51          Min.   : 188.0   Min.   :0.297   Min.   : 85.0
##  Class :character   1st Qu.: 886.5   1st Qu.:3.442   1st Qu.:153.0
##  Mode  :character   Median :1032.0   Median :3.887   Median :174.0
##                     Mean   :1120.0   Mean   :3.778   Mean   :174.2
##                     3rd Qu.:1399.5   3rd Qu.:4.458   3rd Qu.:196.0
##                     Max.   :1786.0   Max.   :5.744   Max.   :234.0
##       AREA            BONUS
##  Min.   : 6.120   Min.   :0.0000
##  1st Qu.: 7.715   1st Qu.:0.0000
##  Median :11.200   Median :1.0000
##  Mean   :13.060   Mean   :0.6078
##  3rd Qu.:15.170   3rd Qu.:1.0000
##  Max.   :40.340   Max.   :1.0000
```

```r
#T-test
T1 <- t.test(profits, alternative="greater", mu=900)
T1
```

```
##
##  One Sample t-test
```

```
##
## data:  profits
## t = 4.3824, df = 50, p-value = 3.011e-05
## alternative hypothesis: true mean is greater than 900
## 95 percent confidence interval:
##  1035.893      Inf
## sample estimates:
## mean of x
##  1120.039
```

```
#Correlation of profits and population
COR = cor(profits, population)
COR
```

```
## [1] 0.6017151
```

```
#Simple linear regression
LM <- lm(profits~population)
LM
```

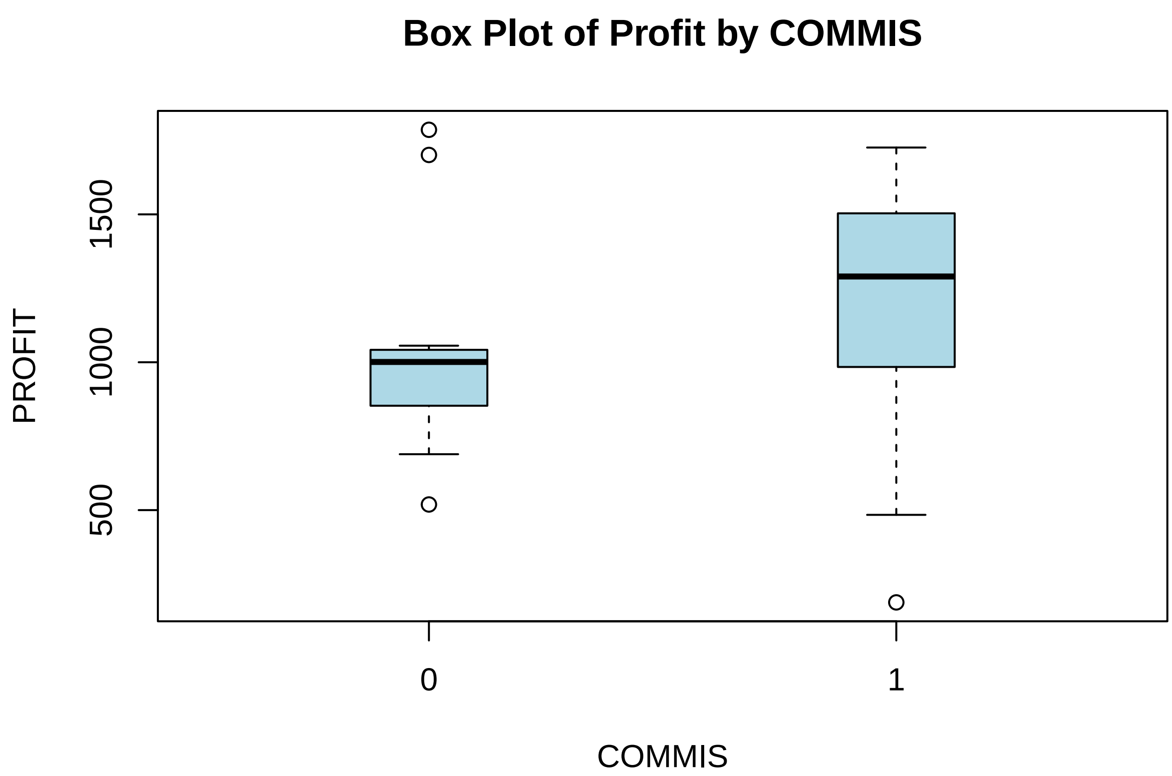
```
##
## Call:
## lm(formula = profits ~ population)
##
## Coefficients:
## (Intercept)   population
##       364.7        199.9
```

```
LM$effects
```

```
##  (Intercept)   population
## -7998.679896  1525.625607   453.296115   288.003190  -104.977253   236.493200
##
##   504.563835    77.212058   303.741605   161.759037  -264.218858    -7.914393
##
##  -120.906618   -27.940263  -357.111478  -364.172632  -145.388611   282.095508
##
##  -546.036834   110.841698  -329.445823   291.730309  -195.332307   435.871091
##
##  -110.183995  -160.837867  -102.803003  -130.248981    26.323935  -138.635554
##
##  -422.802095   369.386972   128.994507   -72.661735  -162.270178   319.236290
##
##   429.636595   335.068176   541.030278  -224.703885  -271.160915  -325.640781
##
##  -199.096173   188.244064  -469.752171  -361.682201   313.111721   190.504253
##
##   470.831067  -106.728847  -154.619828
```

```
  # if multivariables: LM <- lm(y~x1+x2)

#Comparing 2 populations using a boxplot
```

```r
COMMIS <- as.numeric(dataframe[,6]) #extract column 6
boxplot(profits~COMMIS, xlab="COMMIS",ylab="PROFIT", boxwex=0.25,
        main="Box Plot of Profit by COMMIS", col="lightblue")
```




```r
#2 sample t-test
T2 = t.test(profits~COMMIS,alternative=c("less"))
T2
```

```
##
##  Welch Two Sample t-test
##
## data:  profits by COMMIS
## t = -2.0857, df = 47.492, p-value = 0.0212
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0
## 95 percent confidence interval:
##       -Inf -38.51399
## sample estimates:
## mean in group 0 mean in group 1
##        1000.400        1197.226
```

```r
#===================== QUESTION #2: =====================#
library(readxl)

data <- read_excel("Salary.xlsx") #import excel data
```

```r
dataframe <- data.frame(data)

#a - Five number summary
salary <- as.numeric(dataframe[,6]) #extract column 6 data, salaries
salary
```

```
##  [1] 36350 35350 28200 26775 33696 28516 24900 31909 31850 32850 27025 24750
## [13] 28200 23712 25748 29342 31114 24742 22906 24450 19175 20525 27959 38045
## [25] 24832 25400 24800 25500 26182 23725 21600 23300 23713 20690 22450 20850
## [37] 18304 17095 16700 17600 18075 18000 20999 17250 16500 16094 16150 15350
## [49] 16244 16686 15000 20300
```

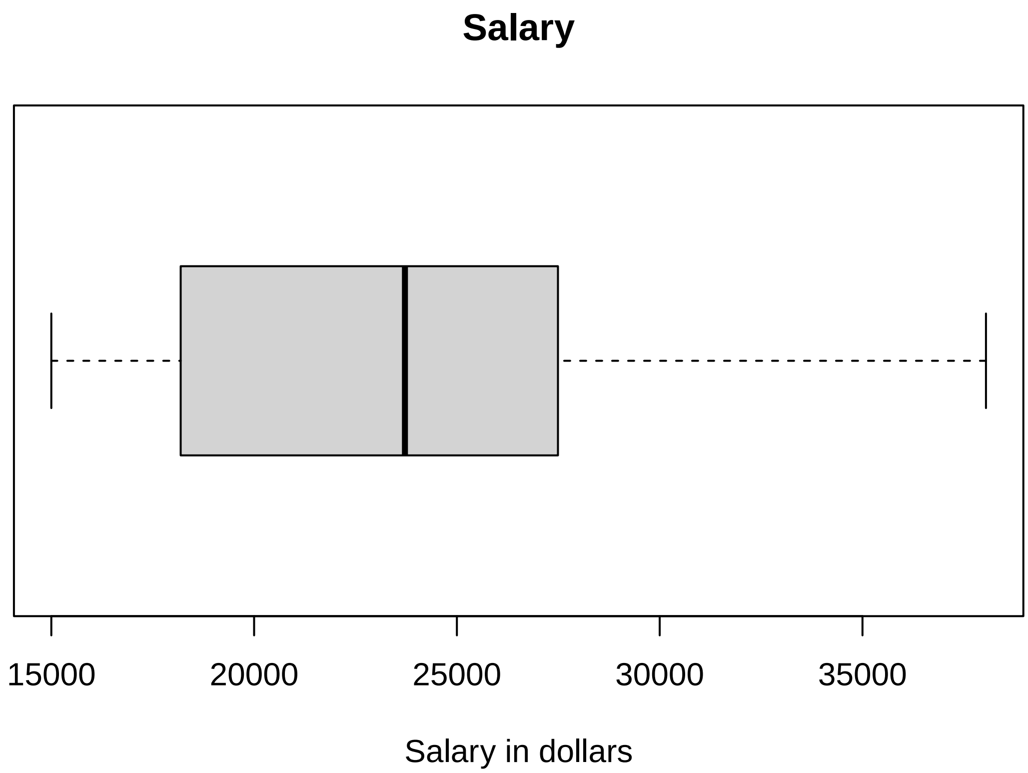
```r
cat("5 number summary = Min, Q1 Median, Median, Q2 Median, Max")
```

```
## 5 number summary = Min, Q1 Median, Median, Q2 Median, Max
```

```r
summary(dataframe)
```

```
##      SEX                RANK               YEARSR          DEGREE         
##  Length:52          Length:52          Min.   : 0.000   Length:52         
##  Class :character   Class :character   1st Qu.: 3.000   Class :character  
##  Mode  :character   Mode  :character   Median : 7.000   Mode  :character  
##                                        Mean   : 7.481                     
##                                        3rd Qu.:11.000                     
##                                        Max.   :25.000                     
##      YEARSD          SALARY     
##  Min.   : 1.00   Min.   :15000  
##  1st Qu.: 6.75   1st Qu.:18247  
##  Median :15.50   Median :23719  
##  Mean   :16.12   Mean   :23798  
##  3rd Qu.:23.25   3rd Qu.:27259  
##  Max.   :35.00   Max.   :38045  
```

```r
boxplot(salary,main="Salary", xlab=
        "Salary in dollars", horizontal=TRUE)
```




```r
#b - Compute the percentages of data points in the intervals mean ± SD,mean ± 2SD.
#What are these percentages expected to be and how close are they
mean <- mean(salary)
sd <- sd(salary)
sd2 <- sd*2

interval1 <- (salary > mean-sd) & (salary < mean+sd)
total <- sum(interval1 == TRUE)
total / length(salary)
```

```
## [1] 0.6346154
```

```r
interval2 <- (salary > mean-sd2) & (salary < mean+sd2)
total <- sum(interval2 == TRUE)
total / length(salary)
```

```
## [1] 0.9615385
```

```r
cat("The emperical rule states that in a Gaussian(normal) distribution, approximately
68% of data falls within 1 standard deviation of the mean, and 95% of the data falls within 2
standard deviations of the mean. We express this using the CLT and generating random numbers
using rnorm")
```

```
## The emperical rule states that in a Gaussian(normal) distribution, approximately
```

```
## 68% of data falls within 1 standard deviation of the mean, and 95% of the data falls within 2
## standard deviations of the mean. We express this using the CLT and generating random numbers
## using rnorm
```

```r
n <- 100000
x <- rnorm(n)
rmean <- mean(x)
rsd <- sd(x)
rsd2 <- 2*rsd

int1 <- (x > rmean-rsd) & (x < rmean+rsd) #mean +- sd
sum(int1 == TRUE) / 100000 #calc percentage
```

```
## [1] 0.68448
```

```r
int2 <- (x > rmean-rsd2) & (x < rmean+rsd2) #mean +- 2*sd
sum(int2 == TRUE) / 100000 #calc percentage
```

```
## [1] 0.95398
```

```r
cat("As seen from these results, since n is extremely large, we can conclude by CLT that ~68.1%
    of data falls between 1 SD of the mean, and ~95.5% of data falls between 2 SDs of the mean.
    Therefore our calculated results from the salary dataset were very close to the expected
    percetanges of normal distribution.")
```

```
## As seen from these results, since n is extremely large, we can conclude by CLT that ~68.1%
##     of data falls between 1 SD of the mean, and ~95.5% of data falls between 2 SDs of the mean.
##     Therefore our calculated results from the salary dataset were very close to the expected
##     percetanges of normal distribution.
```

```r
#c - Repeat part b for a variety of transformations of the salary data. What transformation
# of the data provides a good fit of the data to a Gaussian assumption on the data?

#Take log of all salary datapoints
logsalary <- log(salary)
#Log interval 1
log1 <- (logsalary > mean(logsalary)-sd(logsalary)) & (logsalary < mean(logsalary)+sd(logsalary))
sum(log1 == TRUE) / length(logsalary)
```

```
## [1] 0.6153846
```

```r
#Log interval 2
log2 <- (logsalary > mean(logsalary)-(2*sd(logsalary))) & (logsalary < mean(logsalary)+(2*sd(logsalary)
sum(log2 == TRUE) / length(logsalary)
```

```
## [1] 0.9807692
```

```r
#Take square root of all salary datapoints
rootsalary <- sqrt(salary)
#Root interval 1
root1 <- (rootsalary > mean(rootsalary)-sd(rootsalary)) & (rootsalary < mean(rootsalary)+sd(rootsalary)
sum(root1 == TRUE) / length(rootsalary)
```

```
## [1] 0.6346154
```

```r
#Root interval 2
root2 <- (rootsalary > mean(rootsalary)-(2*sd(rootsalary))) & (rootsalary < mean(rootsalary)+
(2*sd(rootsalary)))
sum(root2 == TRUE) / length(rootsalary)
```

```
## [1] 0.9807692
```

```r
#Z standardize every datapoint in the vector such that mean=0, SD=1
stdsalary <- (salary - mean(salary))/sd(salary)
stdmean <- mean(stdsalary)
stdsd <- sd(stdsalary)
stdsd2 <- 2*stdsd
#Standardize interval 1
std1 <- (stdsalary > stdmean-stdsd) & (stdsalary < stdmean+stdsd)
sum(std1 == TRUE) / length(stdsalary)
```

```
## [1] 0.6346154
```

```r
#Standardize interval 2
std2 <- (stdsalary > stdmean-stdsd2) & (stdsalary < stdmean+stdsd2)
sum(std2 == TRUE) / length(stdsalary)
```

```
## [1] 0.9615385
```

```r
cat("As seen from the 3 transformations and their calculated percentages,
    the best transformation is to Z standardize the datapoints. The standardization produced
    63.4% of datapoints falling within 1 SD from the mean, and 96.2% of datapoints falling
    within 2 SDs from the mean, the closest overall to the expected 68.1% and 95.5%
    for a Gaussian distribution. The log and root transformations had 61.5% & 98.1%, and
    63.5% & 98.1%, respectively.")
```

```
## As seen from the 3 transformations and their calculated percentages,
##     the best transformation is to Z standardize the datapoints. The standardization produced
##     63.4% of datapoints falling within 1 SD from the mean, and 96.2% of datapoints falling
##     within 2 SDs from the mean, the closest overall to the expected 68.1% and 95.5%
##     for a Gaussian distribution. The log and root transformations had 61.5% & 98.1%, and
##     63.5% & 98.1%, respectively.
```