CS544 Final

Senhao Li

```
library(ggplot2)
library(sampling)
```

I. Data Preparation:

1. Download the original data from Kaggle:https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

This is a fictional data created by IBM scientists to explore the factors that cause employees' attrition, and to provide employers strategies in human resource.

It contains a total number of 1,470 employees, and 35 variables for each of them.

2. Read the original csv file into R:

```
attrition <- read.csv("Attrition.csv")
```

3. Save the environment data as Rdata file for future convinience:

```
save(attrition, file="Attrition.Rdata")
```

4. Load the Rdata file into R:

```
load("Attrition.Rdata")
```

II. Data Analysis:

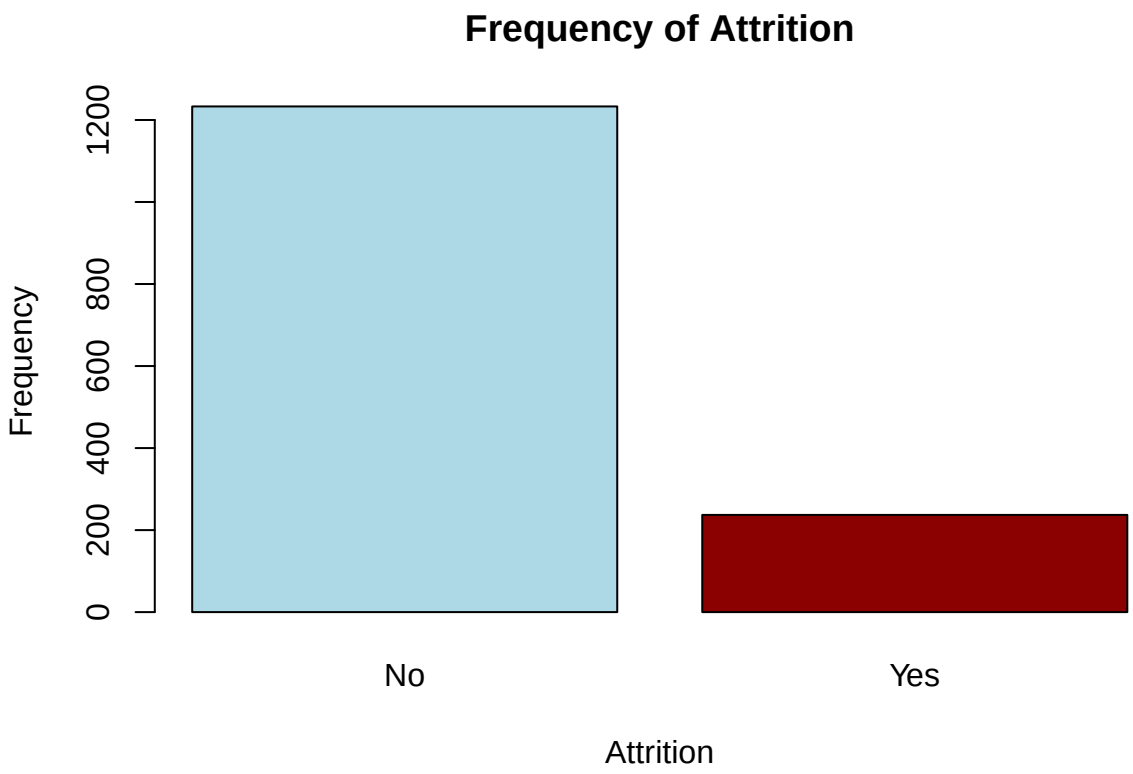
1. Do the analysis as in Module3 for categorical and numerical data. Show appropriate plots for your data:

1.1 Analysis on Categorical variables:

1.1.1 Show the frequencies of attrition and non-attrition employees:

Among 1,470 employees, 237 of them leave the company.

```
barplot(table(attrition$Attrition), col=c("lightblue","darkred"),
        main="Frequency of Attrition",ylab="Frequency",xlab="Attrition")
```

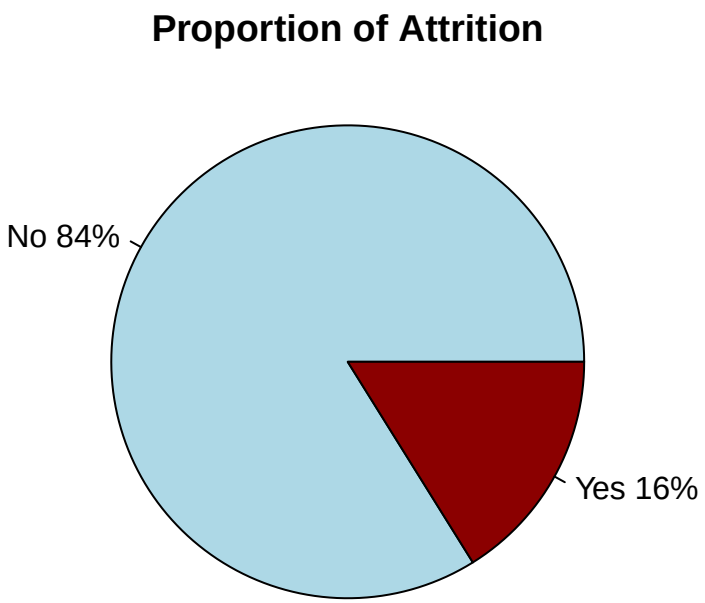



```
data <- table(attrition$Attrition)
```

1.1.2 Show the proportions of attrition and non-attrition employees:

The attrition group of employees account for 16% of the total.

```
slice.labels <- names(data)
slice.percents <- round(data/sum(data)*100)
slice.labels <- paste(slice.labels, slice.percents)
slice.labels <- paste(slice.labels, "%", sep="")

pie(table(attrition$Attrition)/length(attrition$Attrition),
    col=c("lightblue","darkred"),
    radius=1,
    labels=slice.labels,
    main="Proportion of Attrition")
```



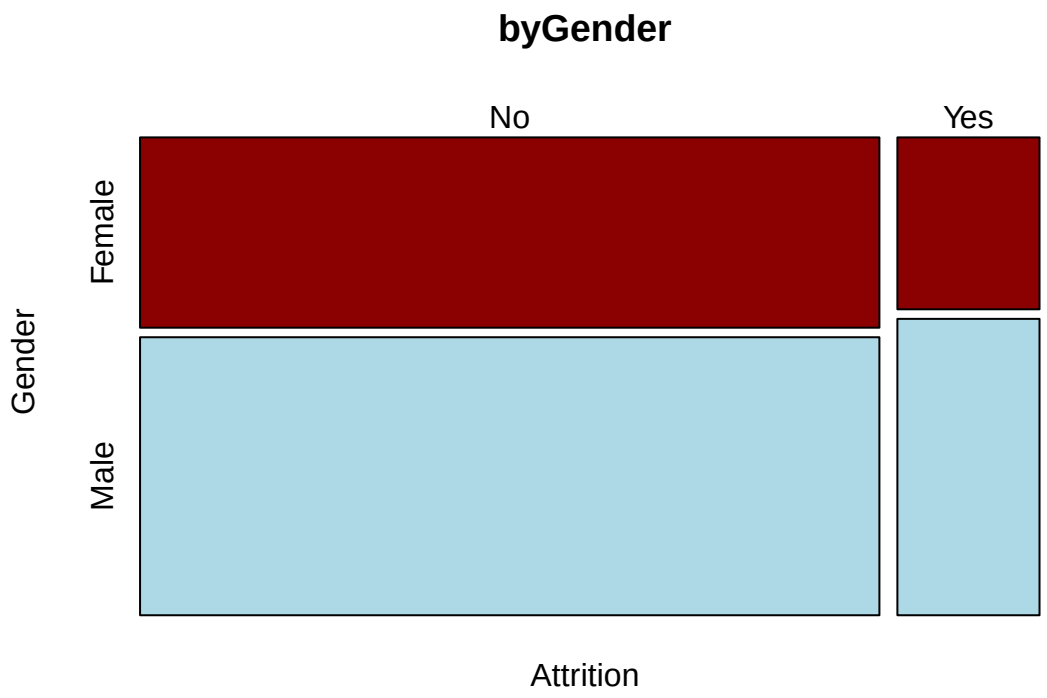
1.1.3 Show the mosaic plot to compare the attrition proportion within different variables:

```
attrition_tb <- table(attrition[,c(2,3,5,7,12,18,23)])
```

Attrition proportion by gender mosaic plot:

There is a slightly higher proportion of attrition in male employees.

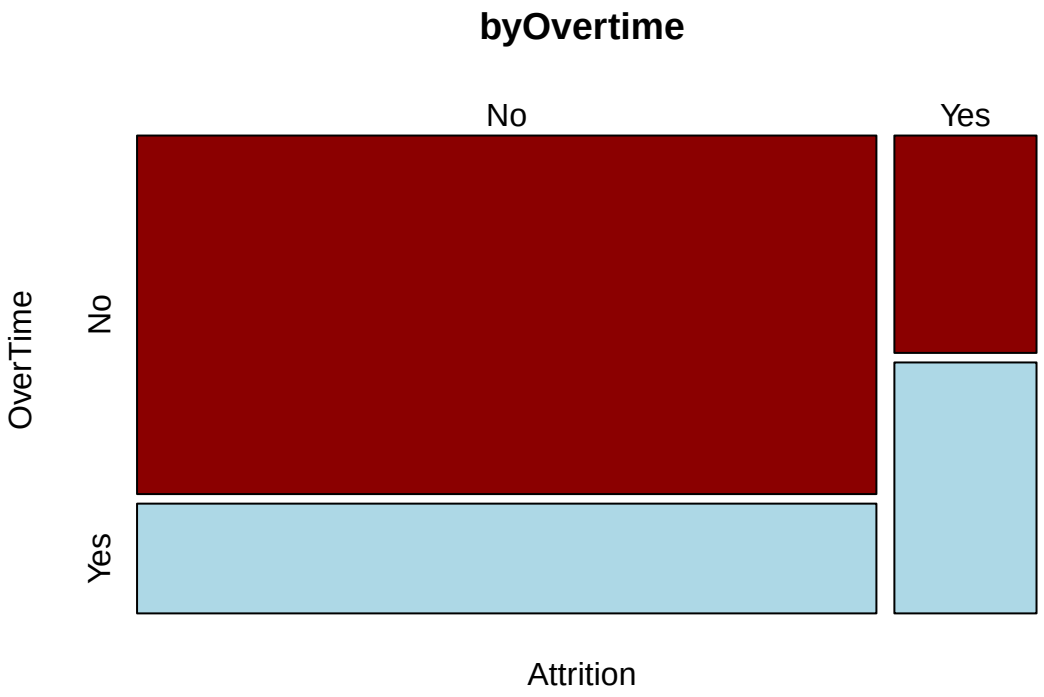
```
byGender <- margin.table(attrition_tb, c(1,5))
mosaicplot(byGender, col=c("darkred","lightblue"), cex.axis =1)
```



Attrition proportion by overtime mosaic plot:

There is an obviously higher proportion of attrition within the group who work over time.

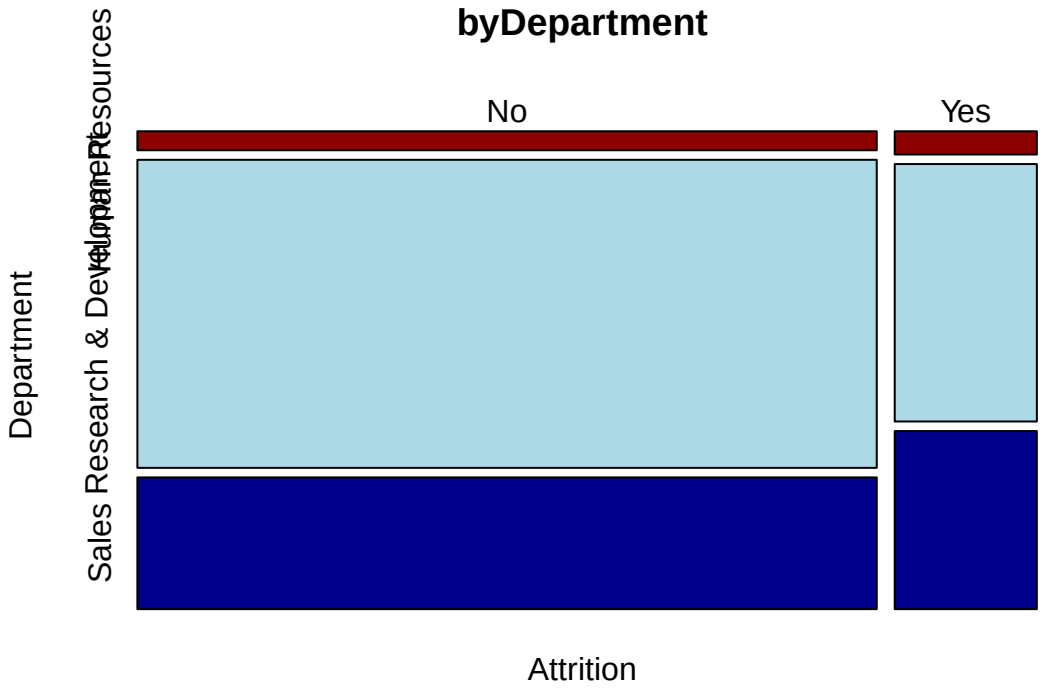
```
byOvertime <- margin.table(attrition_tb,c(1,7))
mosaicplot(byOvertime, col=c("darkred","lightblue"), cex.axis =1)
```



Attrition proportion by department mosaic plot:

The sales department has the highest proportion of attrition, followed by HR department.

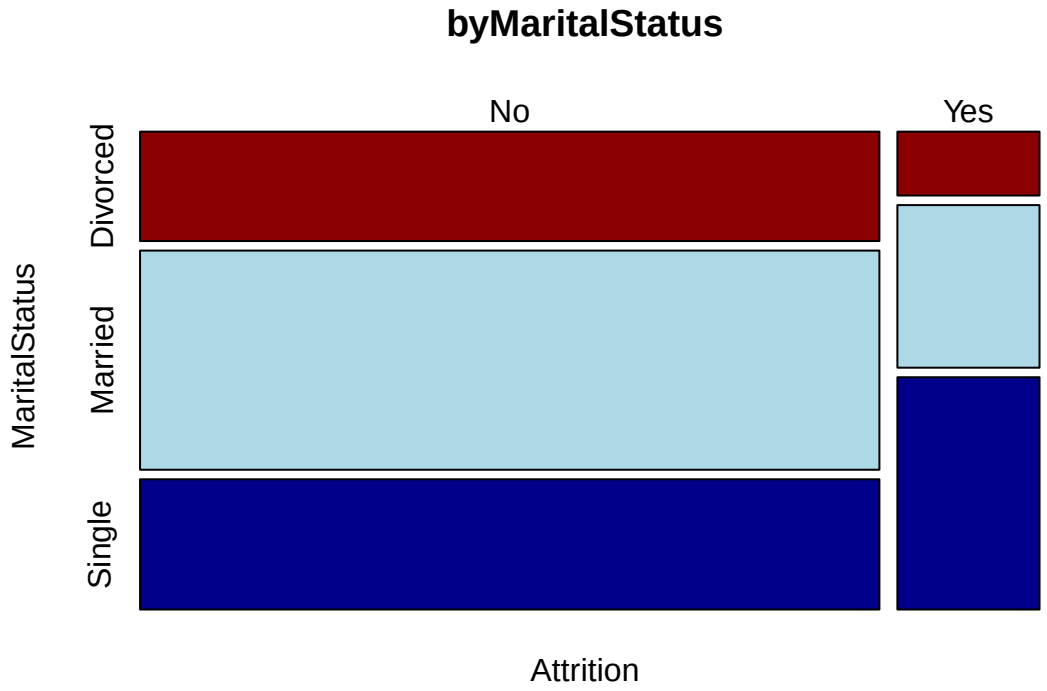
```
byDepartment <- margin.table(attrition_tb,c(1,3))
mosaicplot(byDepartment, col=c("darkred","lightblue","darkblue"), cex.axis =1)
```



Attrition proportion by MaritalStatus mosaic plot:

There is an obviously higher proportion of attrition within the single group;

```
byMaritalStatus <- margin.table(attrition_tb, c(1,6))
mosaicplot(byMaritalStatus, col=c("darkred","lightblue","darkblue"), cex.axis =1)
```



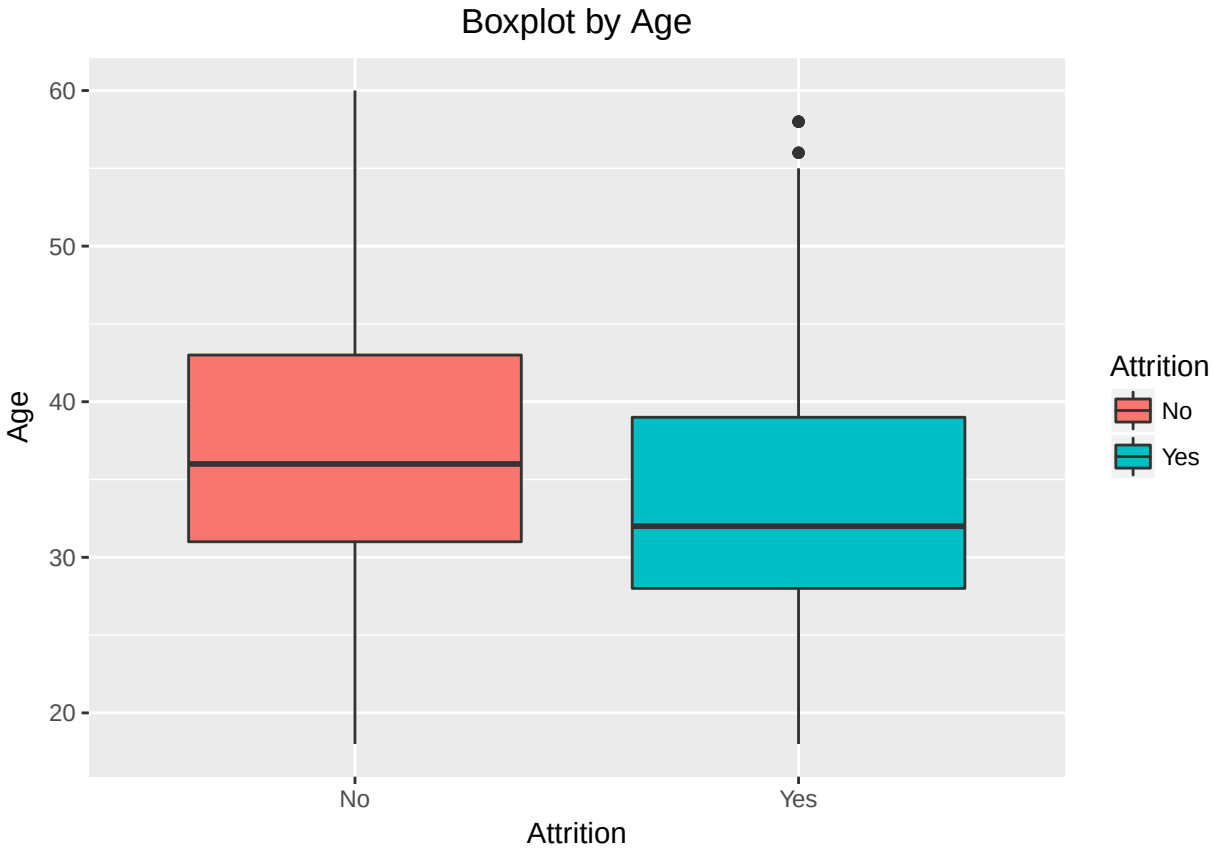
1.2 Analysis on Numeric variables:

1.2.1 Boxplot comparison:

Age comparison between attrition and non-attrition group:

Except two outliers within the Attrition group, those who leave the company are generally younger than those who stay.

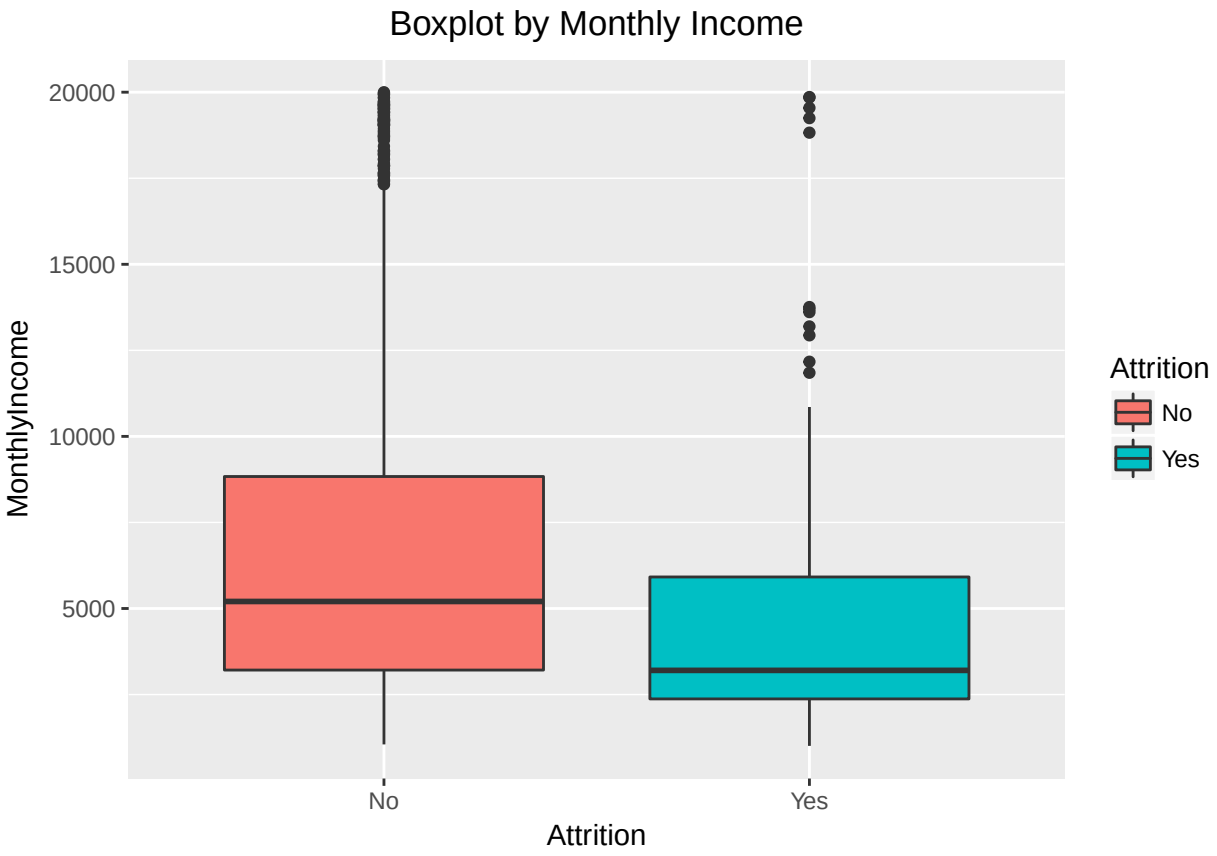
```
ggplot(data = attrition, aes(x=Attrition, y=Age,fill=Attrition)) + geom_boxplot()+ggtitle("Boxplot by Ag
  theme(plot.title = element_text(hjust = .5))
```



Monthly income comparison between attrition and non-attrition group:

Generally, the employees within attrition group gain less salaries except some outliers.

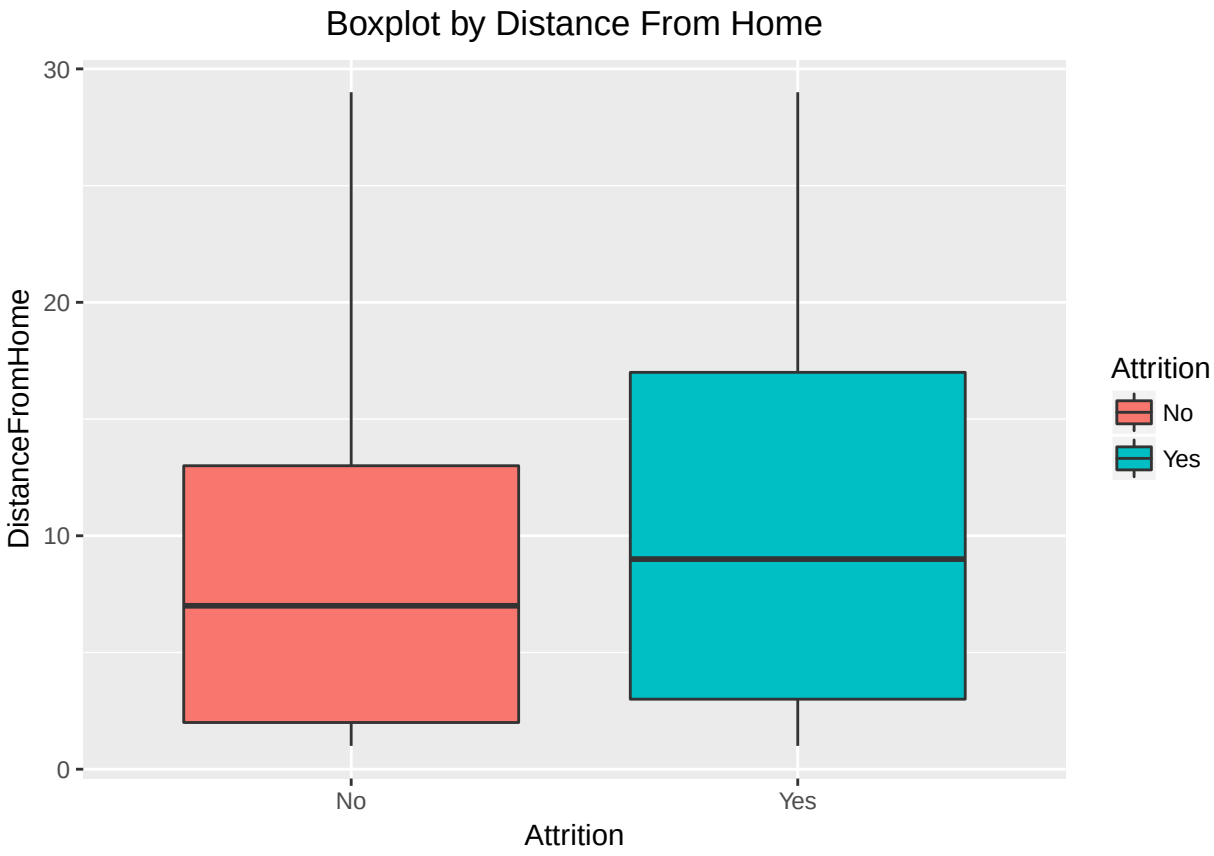
```
ggplot(data = attrition, aes(x=Attrition, y=MonthlyIncome,fill=Attrition)) + geom_boxplot()+ggtitle("Bo
  theme(plot.title = element_text(hjust = .5))
```



Distance-from-home comparison between attrition and non-attrition group:

Generally, Employees who leave the company live farther than those who stay.

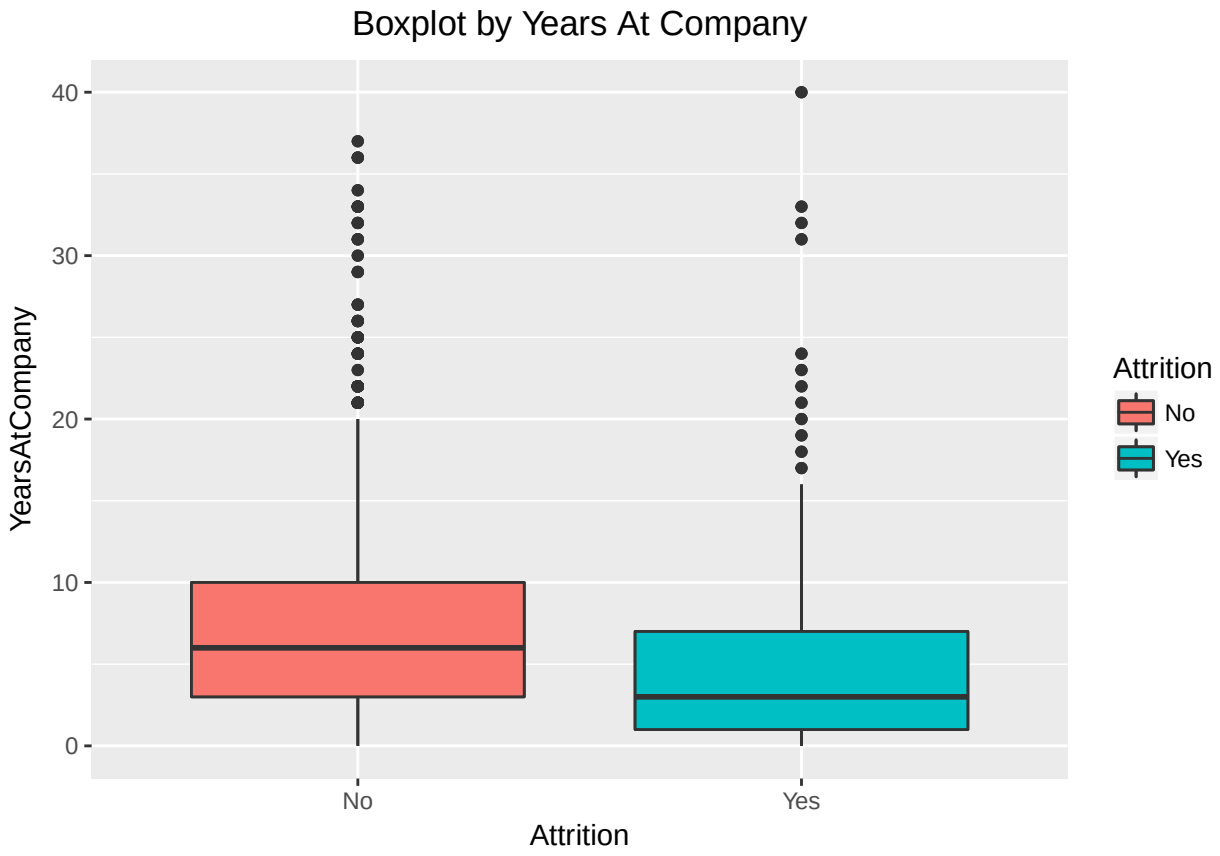
```
ggplot(data = attrition, aes(x=Attrition, y=DistanceFromHome,fill=Attrition)) + geom_boxplot()+ggtitle(
  theme(plot.title = element_text(hjust = .5))
```



Years-at-company comparison between attrition and non-attrition group:

Except some outliers, those who leave the company generally have worked for less time at this company.

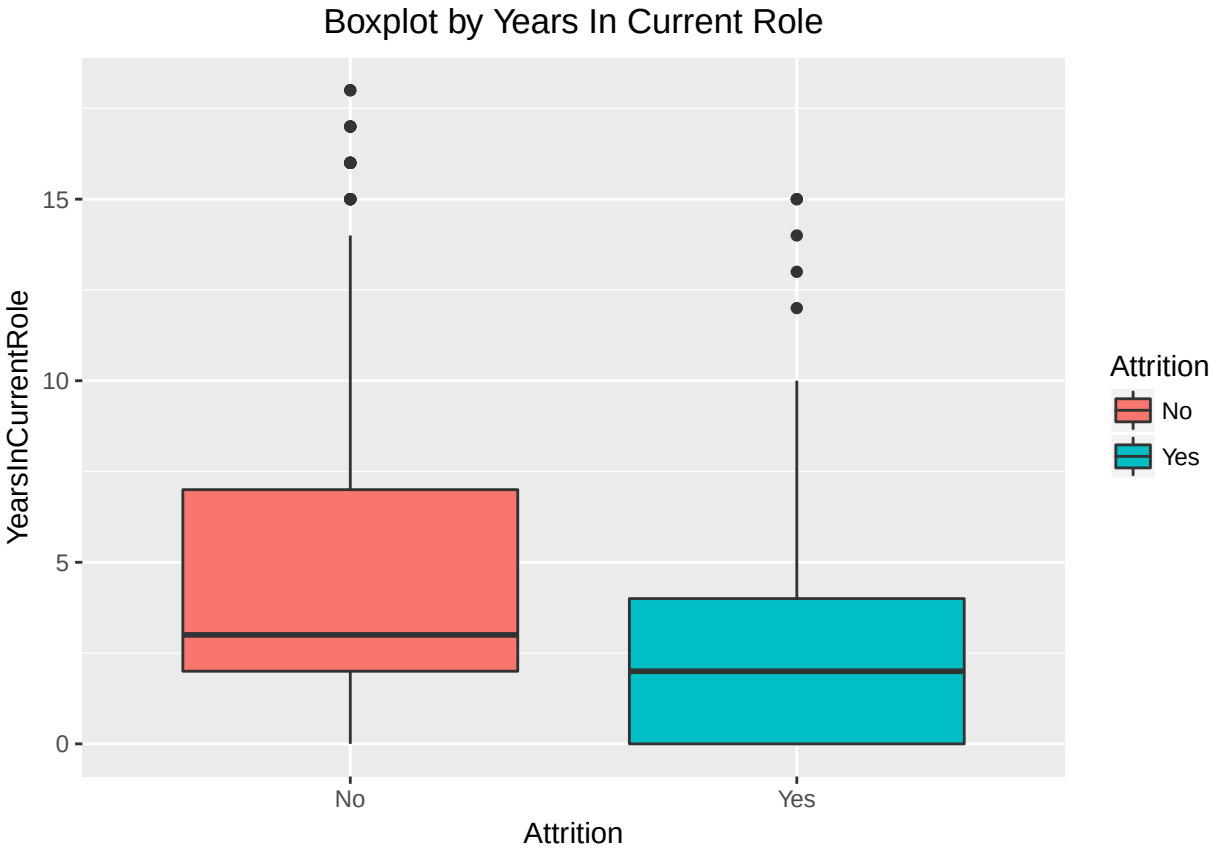
```
ggplot(data = attrition, aes(x=Attrition, y=YearsAtCompany,fill=Attrition)) + geom_boxplot()+ geom_boxpl
  theme(plot.title = element_text(hjust = .5))
```



Years-in-current-role comparison between attrition and non-attrition group:

Except some outliers, those who leave the company generally have worked for less time in their current roles.

```
ggplot(data = attrition, aes(x=Attrition, y=YearsInCurrentRole,fill=Attrition)) + geom_boxplot()+ggtitl
  theme(plot.title = element_text(hjust = .5))
```

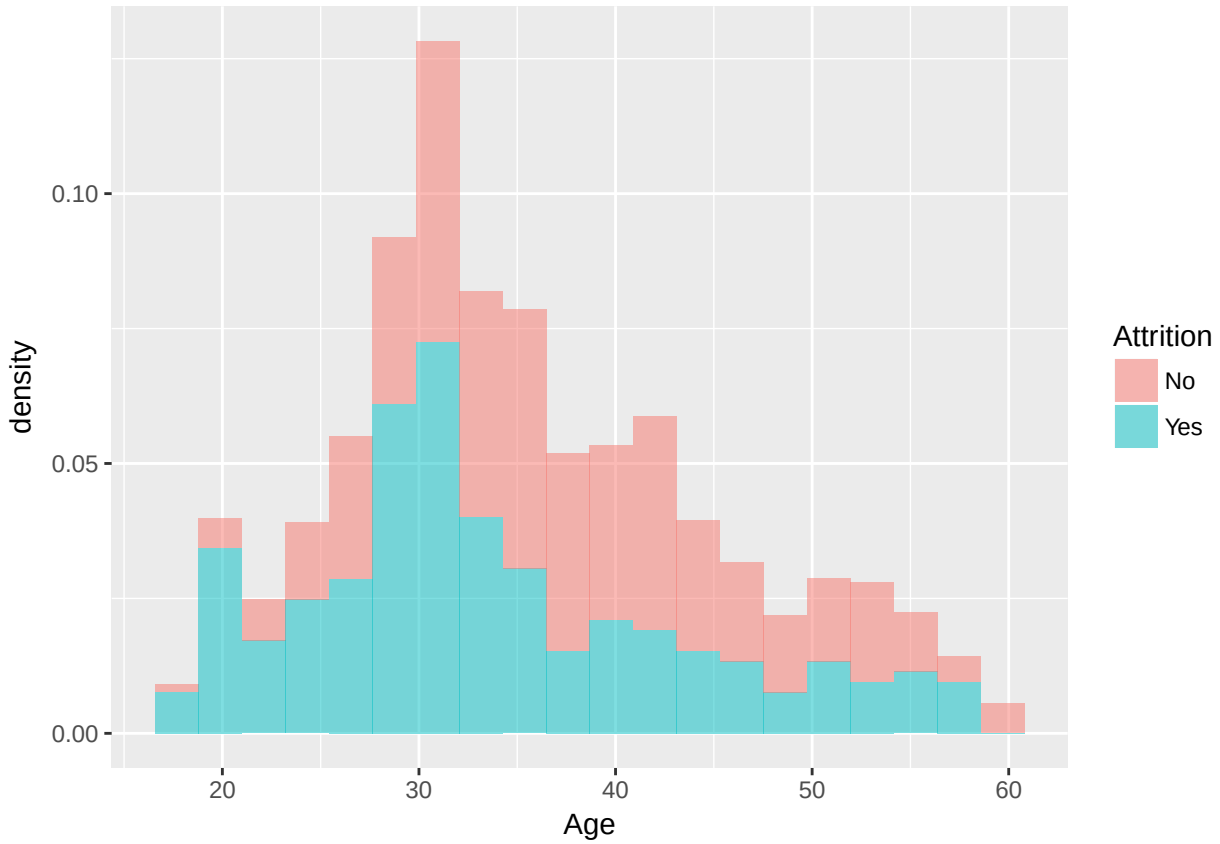


1.2.2 Histogram and distribution comparison:

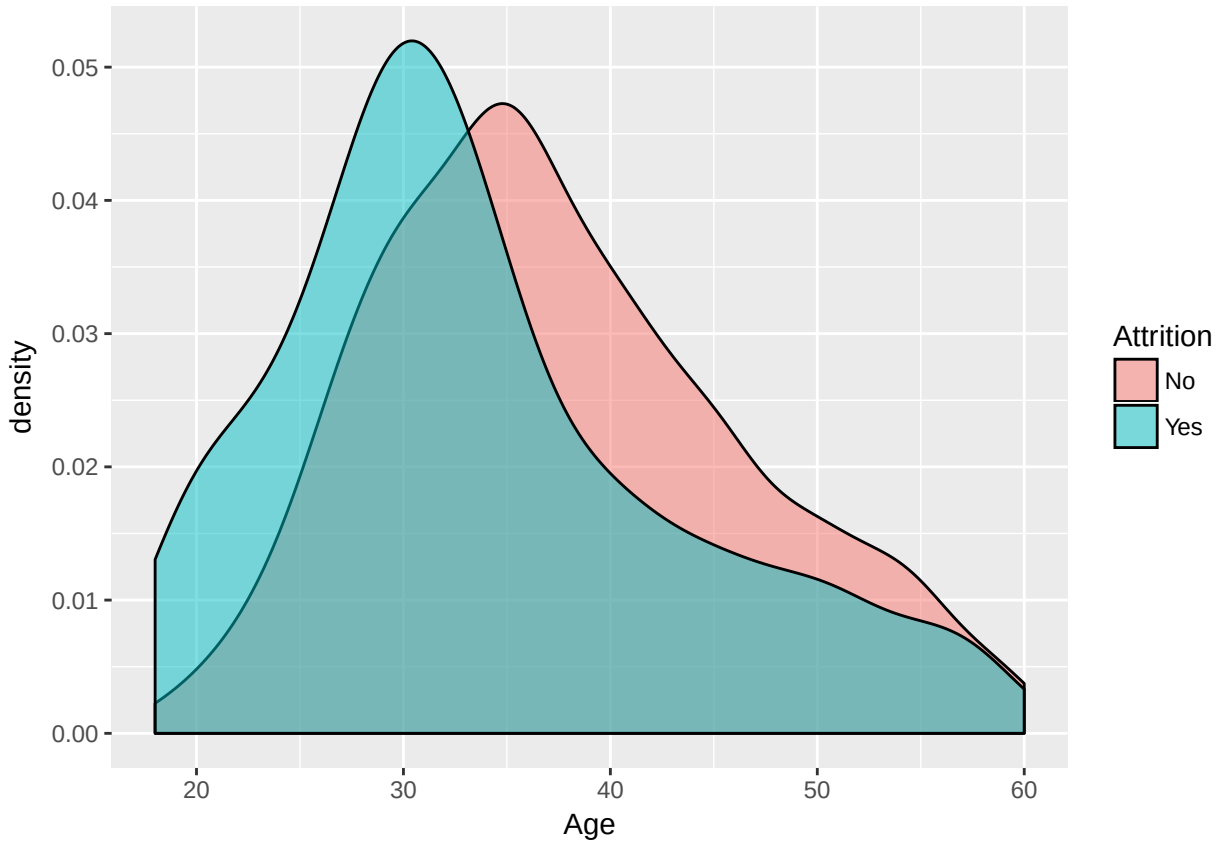
Age comparison between attrition and non-attrition group:

The Attrition group's age distribution is more right-skewed.

```
ggplot(data = attrition, aes(x=Age,fill=Attrition)) + geom_histogram(aes(y=..density..),alpha = 0.5,bins
```



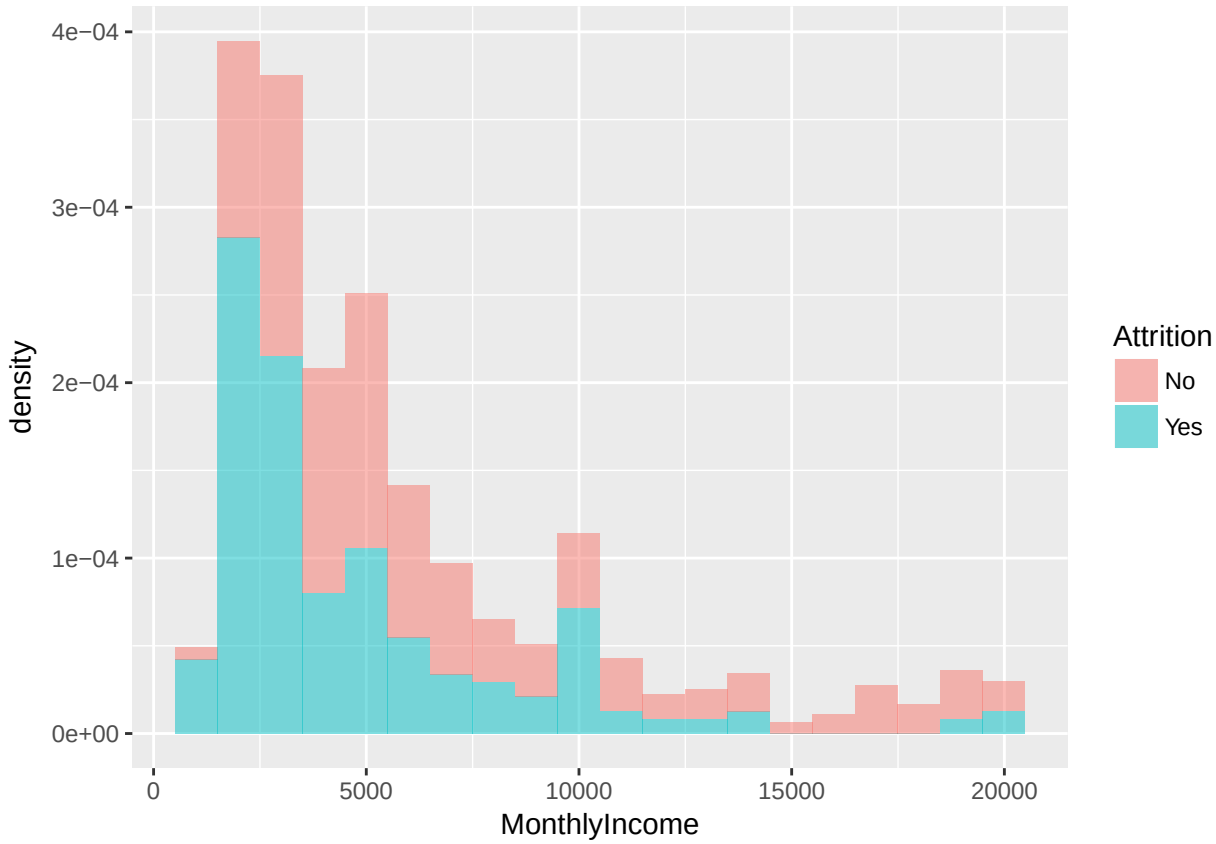

```
ggplot(data = attrition, aes(x=Age,fill=Attrition)) + geom_density(alpha = 0.5)
```



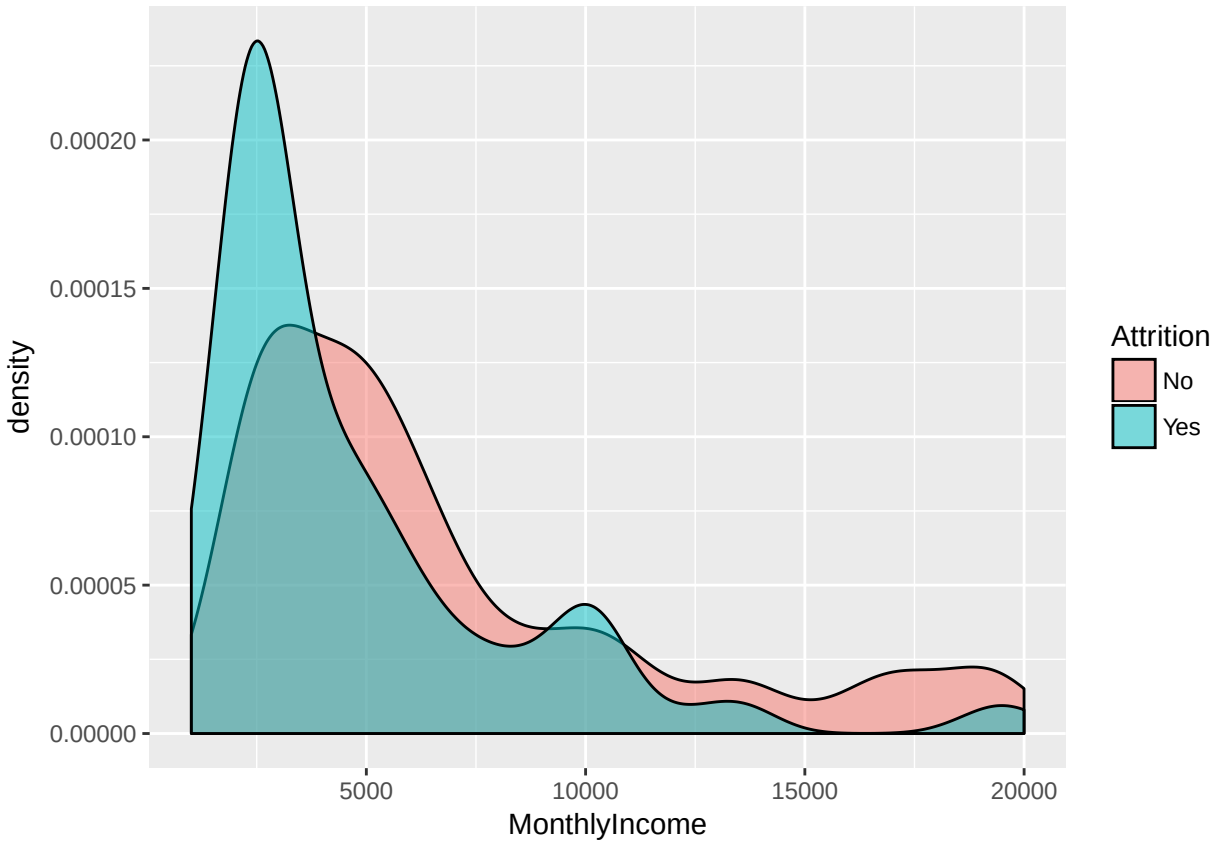
Monthly income comparison between attrition and non-attrition group:

The Attrition group's monthly income distribution is more right-skewed, and more concentrated around $2,500.

```
ggplot(data = attrition, aes(x=MonthlyIncome,fill=Attrition)) + geom_histogram(aes(y=..density..),alpha
```




```
ggplot(data = attrition, aes(x=MonthlyIncome,fill=Attrition)) + geom_density(alpha = 0.5)
```



Years-at-company comparison between attrition and non-attrition group:

The distribution of years that Attrition group have worked at the company is more right-skewed, and more concentrated around 2-3 years.

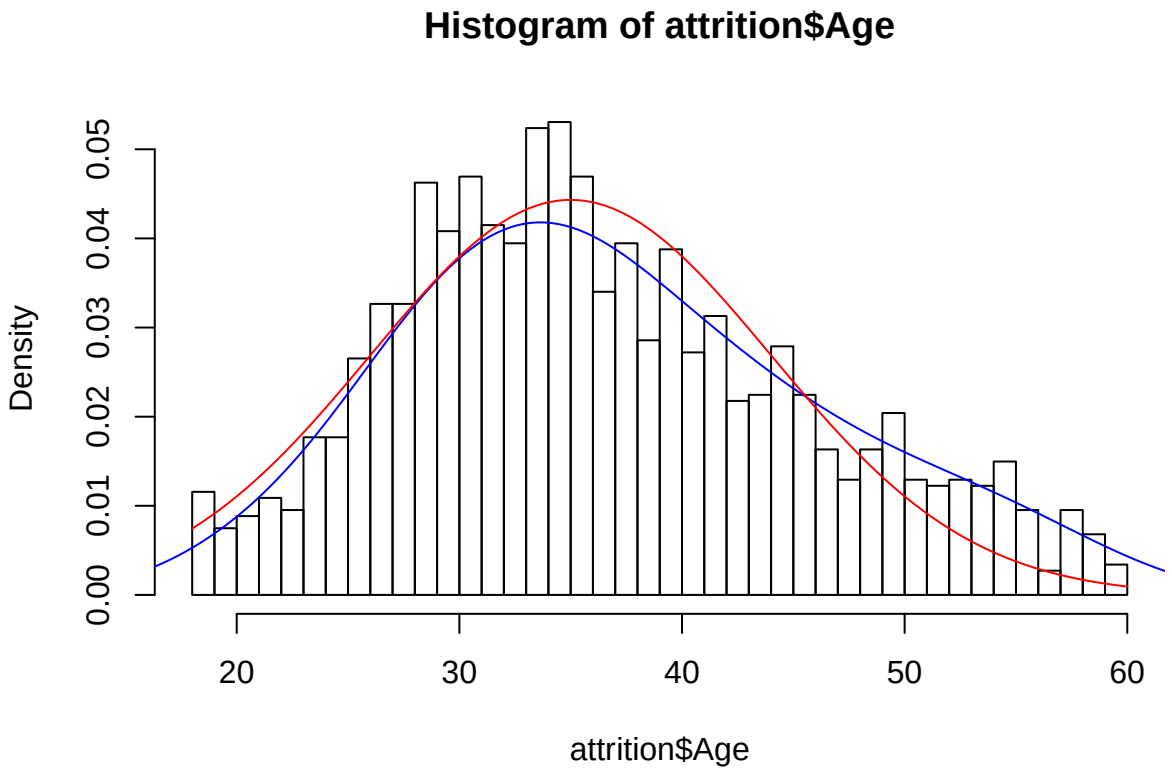
```
ggplot(data = attrition, aes(x=YearsAtCompany,fill=Attrition)) + geom_histogram(aes(y=..density..),alpha
```


```
ggplot(data = attrition, aes(x=YearsAtCompany,fill=Attrition)) + geom_density(alpha = 0.5)
```

2. Pick one variable with numerical data and examine the distribution of the data.

From the graph below, we can see that the distribution of all employees' ages is closer to a normal distribution with a slightly right-skewed tendency.

```
hist(attrition$Age, prob=T, breaks = 50)
lines(density(attrition$Age, adjust=2), col="blue") # add a smooth density line
curve(dnorm(x, mean = 35, sd = 9), add = TRUE, col = "red") # add a normal distribution curve
```



3. Draw various random samples of the data and show the applicability of the Central Limit Theorem for this variable.

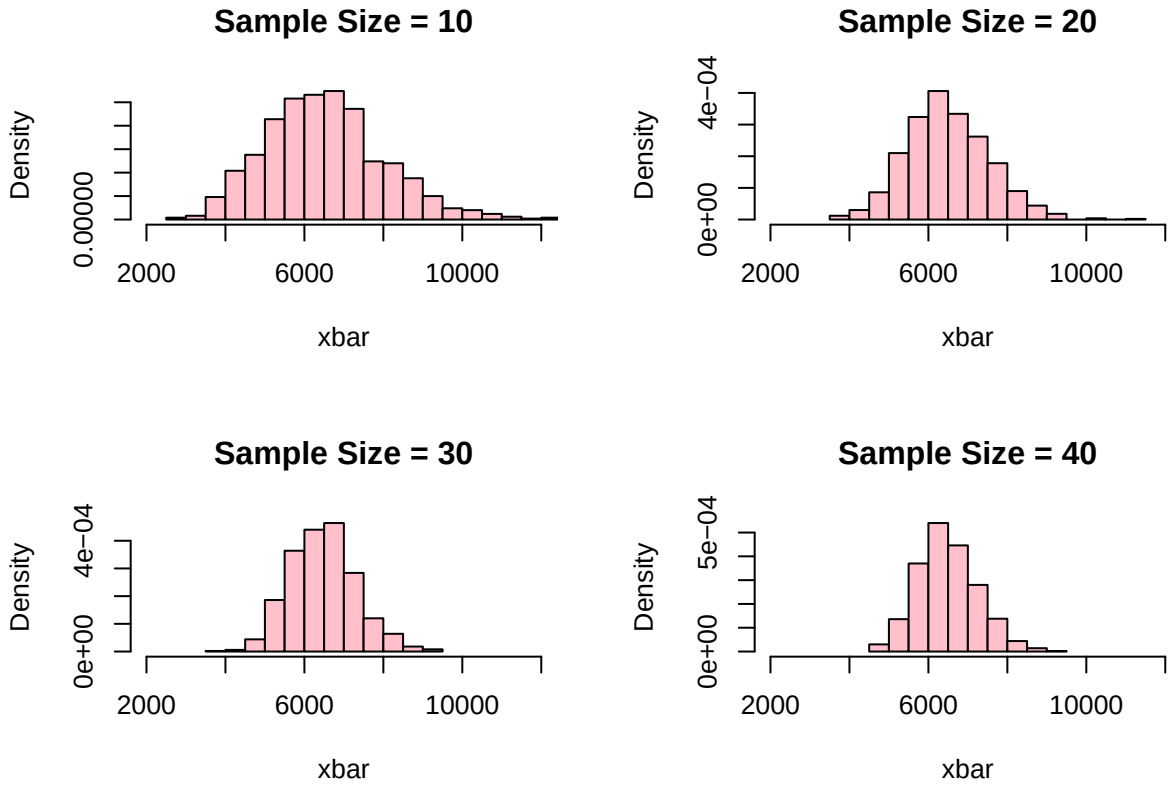
I drew 1,000 random samples of the employees' Monthly Income without replacement:

The distributions of these samples' means are close to normal distribution; as sample size getting larger, the SDs of these distributions decrease, while their means are the same

```
par(mfrow = c(2,2))
samples <- 1000
xbar <- numeric(samples)
for (size in c(10, 20, 30, 40)) {
  for (i in 1:samples) {
    xbar[i] <- mean(sample(attrition$MonthlyIncome, size = size,
                           replace = TRUE))

  }

  hist(xbar, prob = TRUE,
       breaks = 15,
       main = paste("Sample Size =", size),col="pink",xlim=c(2000,12000))

  cat("Sample Size = ", size, " Mean = ", mean(xbar),
      " SD = ", sd(xbar), "\n")
}
```

```
## Sample Size =  10  Mean =  6513.502  SD =  1506.872
```

```
## Sample Size =  20  Mean =  6484.429  SD =  1046.692
## Sample Size =  30  Mean =  6475.153  SD =  834.6579
```



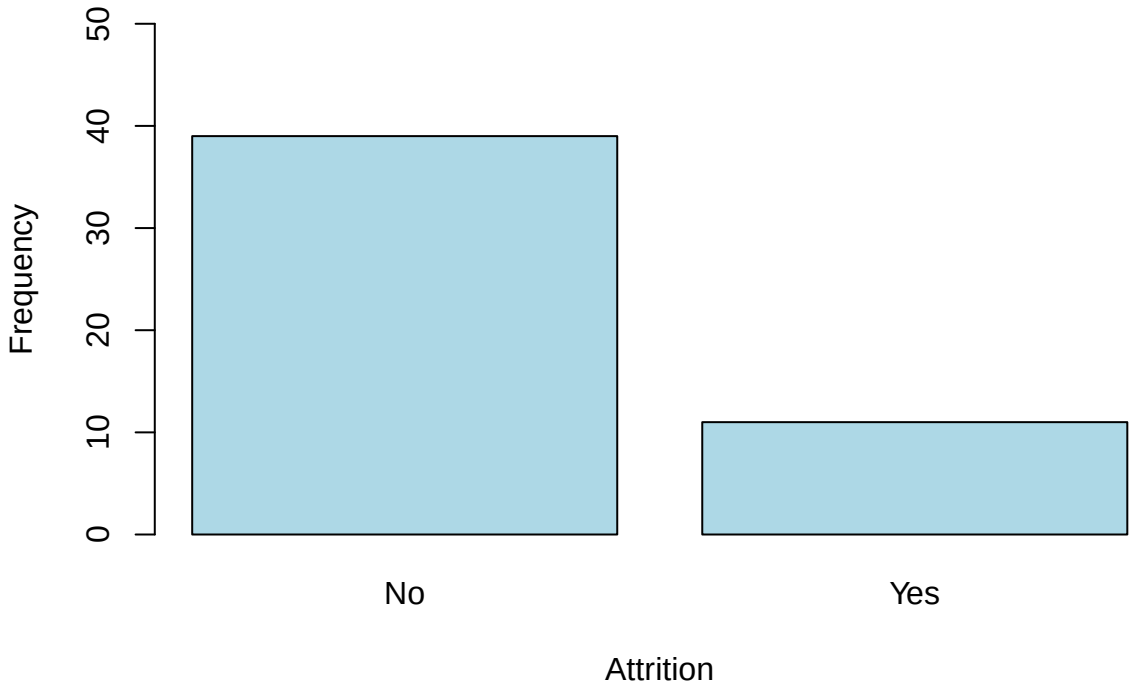

```
## Sample Size =  40  Mean =  6485.606  SD =  749.398
```

```
par(mfrow = c(1,1))
```

4. Show how various sampling methods can be used on your data.

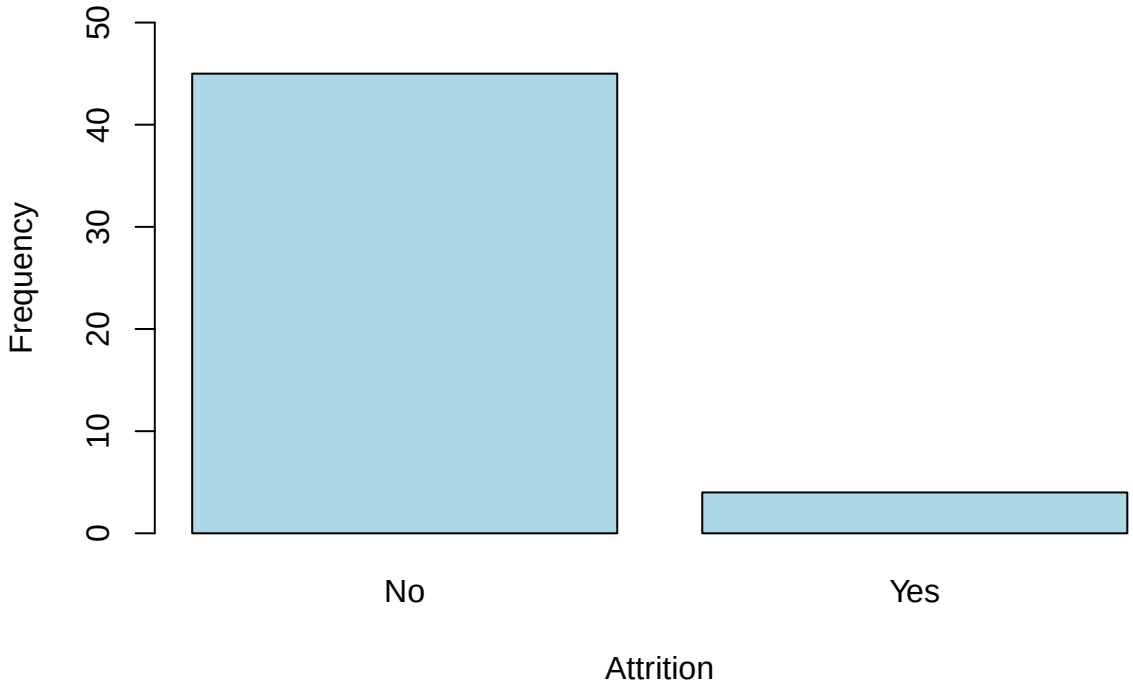
4.1 simple random sample without replacement:

```
set.seed(123)
size <- 50
random_sample <- srswor(size, nrow(attrition))
random_sample_rows <- (1:nrow(attrition))[random_sample!= 0]
random_sample_srswor <- attrition[random_sample_rows,]
barplot(table(random_sample_srswor$Attrition), col="lightblue",
        ylim=c(0,50),xlab="Attrition", ylab="Frequency")
```



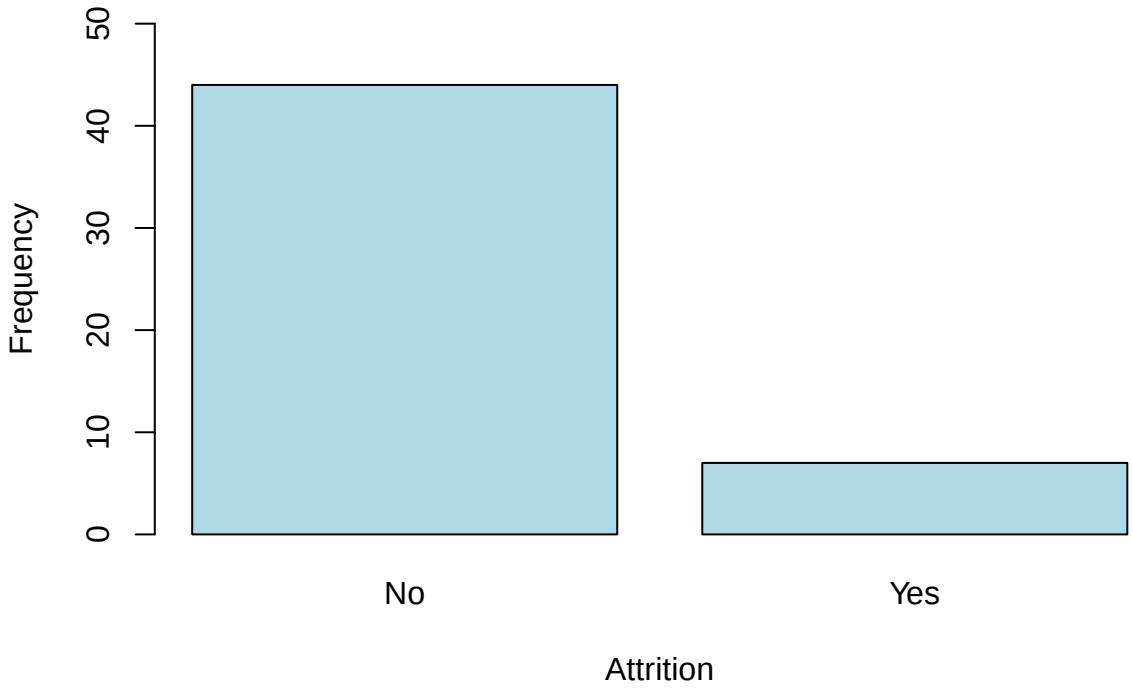
4.2 simple random sample with replacement:

```
random_sample <- srswr(size, nrow(attrition))
random_sample_rows <- (1:nrow(attrition))[random_sample!= 0]
random_sample_srswr <- attrition[random_sample_rows,]
barplot(table(random_sample_srswr$Attrition), col="lightblue", ylim=c(0,50),
       xlab="Attrition", ylab="Frequency")
```



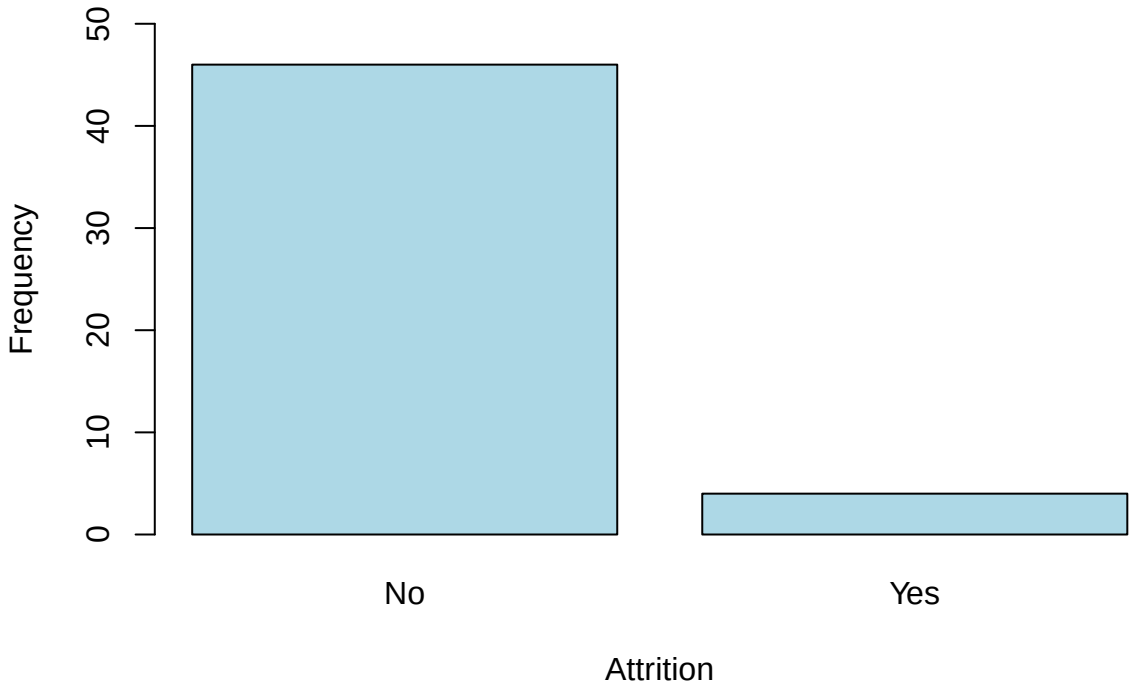
4.3 systematic sampling:

```
numbers_in_each_group <- floor(nrow(attrition)/size)
item_chosen <- sample(numbers_in_each_group, 1)
systematic_sample <- seq(item_chosen, nrow(attrition), by=numbers_in_each_group)
systematic_sample <- attrition[systematic_sample,]
barplot(table(systematic_sample$Attrition), col="lightblue", ylim=c(0,50),
        xlab="Attrition", ylab="Frequency")
```



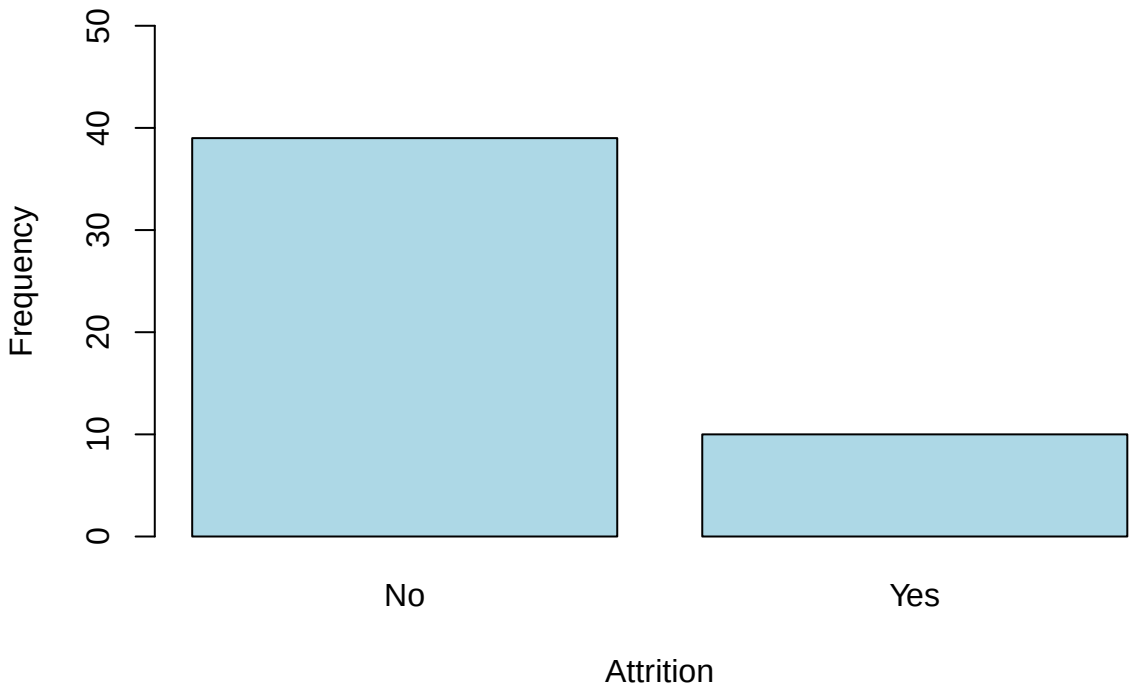
4.4 Unequal Probabilities:

```
inclusion_probabilities <- inclusionprobabilities(attrition$MonthlyIncome,50)

upsystematic_sample <- UPsystematic(inclusion_probabilities)
upsystematic_sample_rows <- (1:nrow(attrition))[upsystematic_sample!= 0]
upsystematic_sample <- attrition[upsystematic_sample_rows,]
barplot(table(upsystematic_sample$Attrition), col="lightblue", ylim=c(0,50),
        xlab="Attrition", ylab="Frequency")
```



4.5 stratified sample

```
order <- order(attrition$JobLevel)
attrition_ordered <- attrition[order,]
freq <- table(attrition$JobLevel)
size <- round(freq/sum(freq)*50)
stratified_sample <- strata(attrition_ordered, stratanames = "JobLevel", size=size, method="srswr")
stratified_sample <- getdata(attrition, stratified_sample)
barplot(table(stratified_sample$Attrition), col="lightblue", ylim=c(0,50),
        xlab="Attrition", ylab="Frequency")
```



5. For confidence levels of 80 and 90, show the confidence intervals of the mean of the numeric variable for various samples and compare against the population mean.

```
options(digits=4)
set.seed(120)

pop.mean <- mean(attrition$MonthlyIncome)
pop.sd <- sd(attrition$MonthlyIncome)

alpha1 <- 1-80/100
z1 <- qnorm(1-alpha1/2)

alpha2 <- 1-90/100
z2 <- qnorm(1-alpha2/2)
```

5.1 comparison of conf interval between different sample size:

As the sample size n increases (in the denominator), the margin of error decreases.

5.1.1 80% confidence level:

```
for (size in c(10, 30, 50, 100)) {
    sd.sample.means_80 <- pop.sd/sqrt(size)
    sample.data_80 <- sample(attrition$MonthlyIncome, size=size)
    xbar <- mean(sample.data_80)
    str <- sprintf("sample size=%2d: 80%% Conf Interval = %.2f - %.2f, margin of error = %.2f", size, xb
                   xbar + z1*sd.sample.means_80, (xbar + z1*sd.sample.means_80-(xbar - z1*sd.sample.mean
```

```
    cat(str, "\n")
}
```

```
## sample size=10: 80% Conf Interval = 4343.24 - 8159.16, margin of error = 1907.96
## sample size=30: 80% Conf Interval = 5499.41 - 7702.53, margin of error = 1101.56
## sample size=50: 80% Conf Interval = 6064.18 - 7770.70, margin of error = 853.26
## sample size=100: 80% Conf Interval = 6041.66 - 7248.36, margin of error = 603.35
```

5.1.2 90% confidence level:

```
for (size in c(10, 30, 50, 100)) {
  sd.sample.means_90 <- pop.sd/sqrt(size)
  sample.data_90 <- sample(attrition$MonthlyIncome, size=size)
  xbar <- mean(sample.data_90)
  str <- sprintf("sample size=%2d: 90%% Conf Interval = %.2f - %.2f, margin of error = %.2f", size,
                 xbar - z2*sd.sample.means_90,
                 xbar + z2*sd.sample.means_90, (xbar + z2*sd.sample.means_90-(xbar - z2*sd.sample.means
  cat(str, "\n")
}
```

```
## sample size=10: 90% Conf Interval = 4996.86 - 9894.54, margin of error = 2448.84
## sample size=30: 90% Conf Interval = 6091.63 - 8919.30, margin of error = 1413.84
## sample size=50: 90% Conf Interval = 4513.87 - 6704.17, margin of error = 1095.15
## sample size=100: 90% Conf Interval = 5444.96 - 6993.74, margin of error = 774.39
```

5.2 comparison between conf interval and pop mean:

As the confidence level increases, the precision also increases. As the confidence level decreases, the precision also decreases, and the number of outside range decreases.

5.2.1 80% confidence level:

```
sample.size <- 30
sd.sample.means <- pop.sd/sqrt(sample.size)
sample.data <- sample(attrition$MonthlyIncome, size=sample.size)
xbar <- mean(sample.data)

cat("80% Conf Interval = ",
    xbar - z1*sd.sample.means, "-",
    xbar + z1*sd.sample.means, "\n",
    "80% precision = ",
    xbar + z1*sd.sample.means- xbar + z1*sd.sample.means)
```

```
## 80% Conf Interval =  5148 - 7351
##  80% precision =  2203
```

```
samples <- 20
for (i in 1: samples) {
  sample.data.1 <- sample(attrition$MonthlyIncome, size=sample.size)
  xbar[i] <- mean(sample.data.1)
  str <- sprintf("%2d: xbar = %.2f, CI = %.2f - %.2f",
                 i, xbar[i], xbar[i] - z1*sd.sample.means,
                 xbar[i] + z1*sd.sample.means)
  cat(str,"\n")
}
```

```
##  1: xbar = 7040.73, CI = 5939.17 - 8142.29
##  2: xbar = 5953.30, CI = 4851.74 - 7054.86
##  3: xbar = 8072.57, CI = 6971.01 - 9174.13
##  4: xbar = 6514.07, CI = 5412.51 - 7615.63
##  5: xbar = 6599.83, CI = 5498.27 - 7701.39
##  6: xbar = 5786.70, CI = 4685.14 - 6888.26
##  7: xbar = 7517.23, CI = 6415.67 - 8618.79
##  8: xbar = 7670.97, CI = 6569.41 - 8772.53
##  9: xbar = 6097.30, CI = 4995.74 - 7198.86
## 10: xbar = 7036.33, CI = 5934.77 - 8137.89
## 11: xbar = 6106.97, CI = 5005.41 - 7208.53
## 12: xbar = 7020.80, CI = 5919.24 - 8122.36
## 13: xbar = 9303.23, CI = 8201.67 - 10404.79
## 14: xbar = 6414.80, CI = 5313.24 - 7516.36
## 15: xbar = 6739.20, CI = 5637.64 - 7840.76
## 16: xbar = 6265.73, CI = 5164.17 - 7367.29
## 17: xbar = 4729.80, CI = 3628.24 - 5831.36
## 18: xbar = 6270.23, CI = 5168.67 - 7371.79
## 19: xbar = 6395.10, CI = 5293.54 - 7496.66
## 20: xbar = 7188.13, CI = 6086.57 - 8289.69
```

```
xbar
```

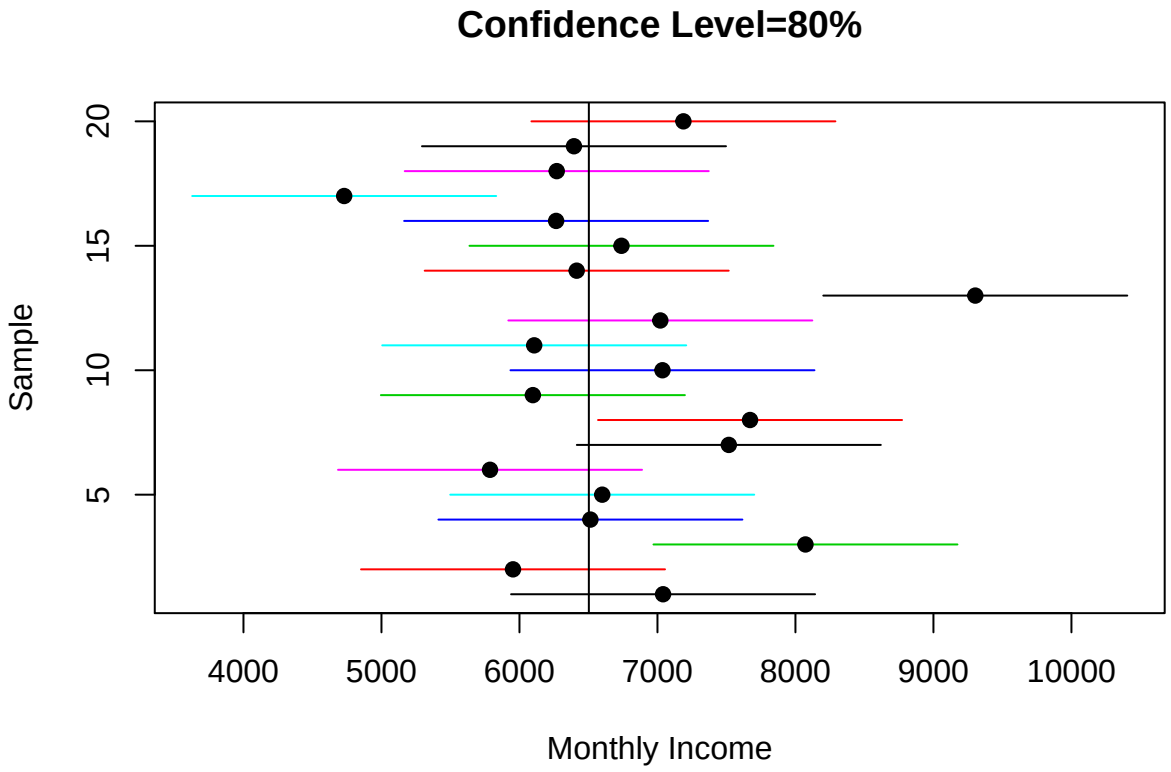
```
##  [1] 7041 5953 8073 6514 6600 5787 7517 7671 6097 7036 6107 7021 9303 6415
## [15] 6739 6266 4730 6270 6395 7188
```

number outside the range

```
sum(abs(xbar-pop.mean) > z1*sd.sample.means)
```

```
## [1] 4
```

```
matplot(rbind(xbar - z1*sd.sample.means, xbar + z1*sd.sample.means),
        rbind(1:samples, 1:samples), type="l",lty=1,ylab="Sample", xlab="Monthly Income", main="Confide
abline(v = pop.mean)
points(xbar,1:samples, pch=19)
```



5.2.2 90% confidence level:

```
set.seed(120)

cat("90% Conf Interval = ",
    xbar - z2*sd.sample.means, "-",
    xbar + z2*sd.sample.means, "\n",
    "90% precision = ",
    xbar + z2*sd.sample.means- xbar + z2*sd.sample.means)
```

```
## 90% Conf Interval =  5627 4539 6659 5100 5186 4373 6103 6257 4683 5622 4693 5607 7889 5001 5325 4852
##  90% precision =  2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 2828 28
```

```
samples <- 20
for (i in 1: samples) {
  sample.data.1 <- sample(attrition$MonthlyIncome, size=sample.size)
  xbar[i] <- mean(sample.data.1)
  str <- sprintf("%2d: xbar = %.2f, CI = %.2f - %.2f",
                 i, xbar[i], xbar[i] - z2*sd.sample.means,
                 xbar[i] + z2*sd.sample.means)
  cat(str,"\n")
}
```

```
##  1: xbar = 6163.47, CI = 4749.63 - 7577.30
##  2: xbar = 4943.53, CI = 3529.70 - 6357.37
##  3: xbar = 6734.47, CI = 5320.63 - 8148.30
##  4: xbar = 6773.03, CI = 5359.20 - 8186.87
##  5: xbar = 6009.37, CI = 4595.53 - 7423.20
##  6: xbar = 4396.20, CI = 2982.36 - 5810.04
##  7: xbar = 5848.67, CI = 4434.83 - 7262.50
```

```
##  8: xbar = 6126.23, CI = 4712.40 - 7540.07
##  9: xbar = 6390.27, CI = 4976.43 - 7804.10
## 10: xbar = 5652.83, CI = 4239.00 - 7066.67
## 11: xbar = 5323.43, CI = 3909.60 - 6737.27
## 12: xbar = 6778.87, CI = 5365.03 - 8192.70
## 13: xbar = 7389.93, CI = 5976.10 - 8803.77
## 14: xbar = 5451.60, CI = 4037.76 - 6865.44
## 15: xbar = 7711.07, CI = 6297.23 - 9124.90
## 16: xbar = 6246.33, CI = 4832.50 - 7660.17
## 17: xbar = 7073.30, CI = 5659.46 - 8487.14
## 18: xbar = 5719.83, CI = 4306.00 - 7133.67
## 19: xbar = 6449.73, CI = 5035.90 - 7863.57
## 20: xbar = 6215.43, CI = 4801.60 - 7629.27
```

```
xbar
```

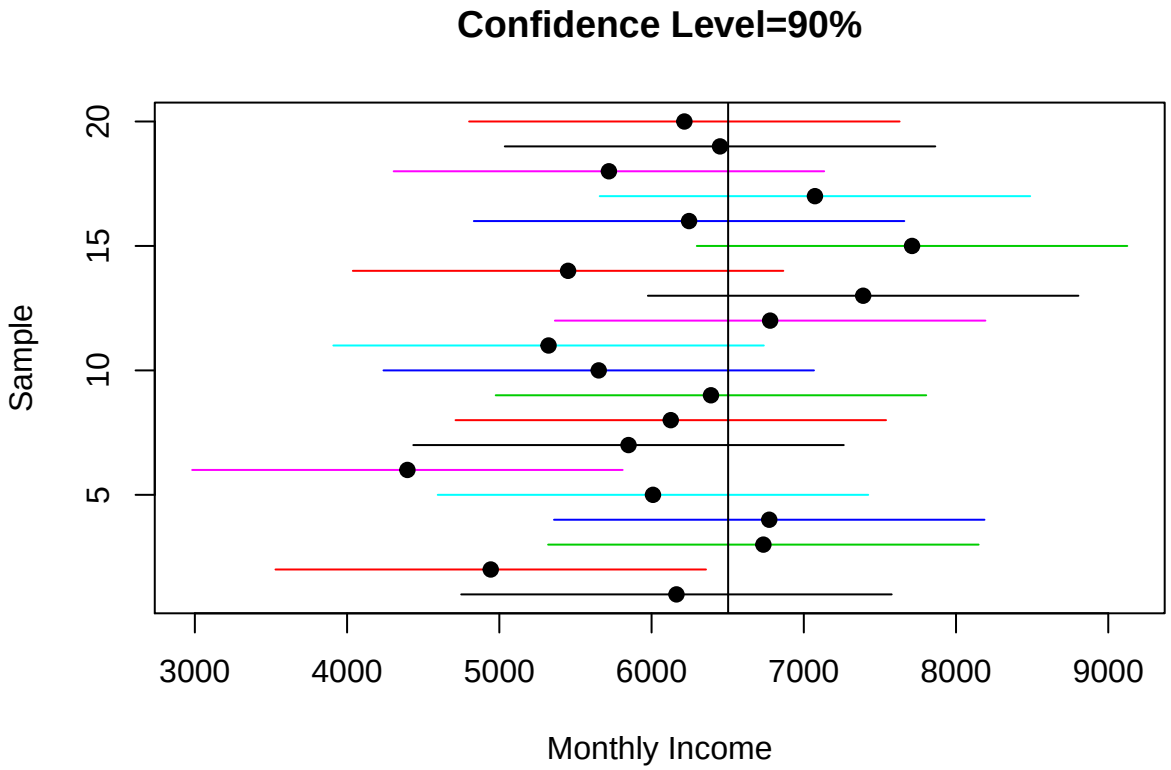
```
##  [1] 6163 4944 6734 6773 6009 4396 5849 6126 6390 5653 5323 6779 7390 5452
## [15] 7711 6246 7073 5720 6450 6215
```

number outside the range

```
sum(abs(xbar-pop.mean) > z2*sd.sample.means)
```

```
## [1] 2
```

```
matplot(rbind(xbar - z2*sd.sample.means, xbar + z2*sd.sample.means),
        rbind(1:samples, 1:samples), type="l",lty=1,ylab="Sample", xlab="Monthly Income", main="Confide
abline(v = pop.mean)
points(xbar,1:samples, pch=19)
```



6. Test of Significance: Critical-Value Approach

```
set.seed(123)

pop.mean <- mean(attrition$MonthlyIncome)
pop.sd <- sd(attrition$MonthlyIncome)

sample.size <- 100
sample.data <- sample(attrition$MonthlyIncome, size=sample.size)

xbar <- mean(sample.data)
xbar
```

```
## [1] 6685
```

```
mu0 <- pop.mean
sigma <- pop.sd
n <- sample.size

z <- (xbar - mu0) / (sigma / sqrt(n))
z
```

```
## [1] 0.3871
```

At a significance level 0.05, the critical values are

```
alpha <- 0.05
c(qnorm(alpha/2), qnorm(1 - alpha/2))
```

```
## [1] -1.96  1.96
```

The test statistic, z, lies in between the critical values. Hence the null hypothesis is not rejected for the given significance level.