STATS_204_HW2

Qi Wang

Question 1:

(a)

```
data.food <- scan(text = "Wendys McDonalds Subway Subway Subway Wendys
Wendys Subway Wendys Subway Subway Subway
Subway Subway Subway", what = "character")
```

(b)

```
table(data.food)
```

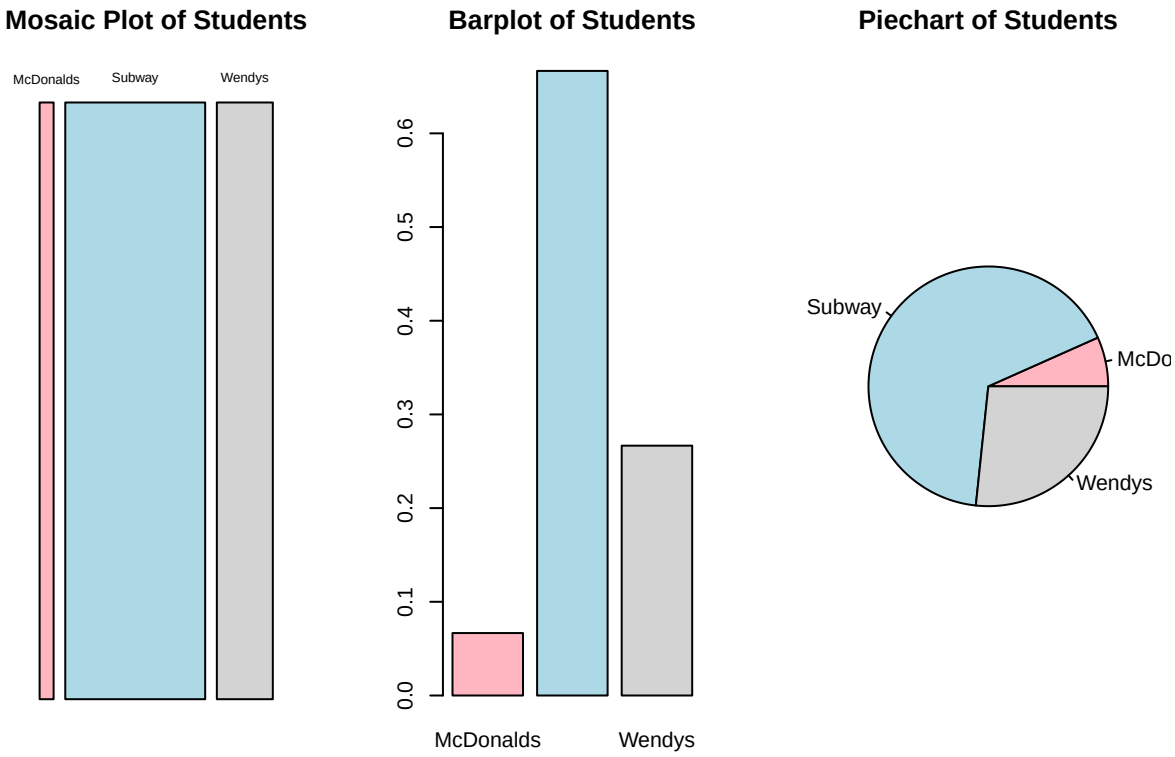
```
## data.food
## McDonalds    Subway    Wendys
##         1        10         4
```

(c)

```
food.table <- table(data.food)/length(data.food)
```

(d)

```
par(mfrow = c(1,3))
mosaicplot(food.table, color = c("lightpink", "lightblue", "lightgrey"),
las = 1, main = "Mosaic Plot of Students", xlab = "")
barplot(food.table, col = c("lightpink", "lightblue", "lightgrey"), main = "Barplot of Students")
pie(food.table, col = c("lightpink", "lightblue", "lightgrey"), radius = 1,
main = "Piechart of Students")
```



Question 2:

```
a <- 0:4
P <- dbinom(a, size = 4, prob = 0.312)
P <- c(P[1:3], P[4] + P[5])
k <- dbinom(a, size = 4, prob = 0.312) * 70
k <- c(k[1:3], k[4] + k[5])
chisq.test(c(17, 31, 17, 5), p = P)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(17, 31, 17, 5)
## X-squared = 0.97692, df = 3, p-value = 0.8068
```

From the result above, we cannot reject the null hypothesis that the data follows a Binomial distribution with size equals 70 and probability of success is 0.312.

```
a <- 0:5
P <- dbinom(a, size = 5, prob = 0.312)
P <- c(P[1:3], P[4] + P[5] + P[6])
k <- dbinom(a, size = 4, prob = 0.312) * 25
k <- c(k[1:3], k[4] + k[5] + k[6])
chisq.test(c(5, 5, 4, 11), p = P)
```

```
## Warning in chisq.test(c(5, 5, 4, 11), p = P): Chi-squared approximation may be
```

```
## incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(5, 5, 4, 11)
## X-squared = 13.359, df = 3, p-value = 0.003922
```

From the chi-square test result above, we can reject the null hypothesis, which means that the data is not following the binomial distribution with size 25 and probability 0.312.

Question 3:

(a)

```
data.twins <- read.table("D:/77/UCSC/study/204/HW/twins.txt", header=TRUE, sep=",",na.strings=".")
data.twins <- na.omit(data.twins)
cate_age <- cut(data.twins$AGE, c(0,30,40,50,100), labels = c("Younger than 30",
"30 - 40", "40 - 50", "Older than 50") )
print(cate_age)
```

```
##   [1] 30 - 40         40 - 50         30 - 40         30 - 40
##   [5] 30 - 40         Younger than 30 40 - 50         Older than 50
##   [9] 30 - 40         40 - 50         Younger than 30 30 - 40
##  [13] 30 - 40         Younger than 30 30 - 40         Younger than 30
##  [17] 30 - 40         Younger than 30 40 - 50         40 - 50
##  [21] Older than 50   Younger than 30 30 - 40         Older than 50
##  [25] 40 - 50         30 - 40         Younger than 30 30 - 40
##  [29] Older than 50   Younger than 30 Younger than 30 30 - 40
##  [33] 40 - 50         Younger than 30 40 - 50         Older than 50
##  [37] 30 - 40         Older than 50   30 - 40         Older than 50
##  [41] Younger than 30 30 - 40         Older than 50   Older than 50
##  [45] 40 - 50         Younger than 30 Younger than 30 Younger than 30
##  [49] Younger than 30 Older than 50   Older than 50   40 - 50
##  [53] 30 - 40         Younger than 30 30 - 40         Younger than 30
##  [57] 30 - 40         30 - 40         30 - 40         30 - 40
##  [61] 30 - 40         30 - 40         30 - 40         30 - 40
##  [65] 30 - 40         30 - 40         40 - 50         30 - 40
##  [69] Younger than 30 40 - 50         Older than 50   Younger than 30
##  [73] 40 - 50         Younger than 30 40 - 50         30 - 40
##  [77] Older than 50   Younger than 30 30 - 40         Older than 50
##  [81] 40 - 50         Younger than 30 Older than 50   40 - 50
##  [85] 30 - 40         40 - 50         30 - 40         30 - 40
##  [89] 30 - 40         30 - 40         30 - 40         30 - 40
##  [93] Younger than 30 40 - 50         30 - 40         Younger than 30
##  [97] 40 - 50         Older than 50   30 - 40         30 - 40
## [101] 30 - 40         30 - 40         40 - 50         40 - 50
## [105] 30 - 40         Younger than 30 Younger than 30 Younger than 30
## [109] 40 - 50         30 - 40         30 - 40         Younger than 30
## [113] 40 - 50         40 - 50         Older than 50   40 - 50
## [117] Younger than 30 30 - 40         30 - 40         Younger than 30
## [121] Older than 50   Younger than 30 Older than 50   Younger than 30
## [125] 30 - 40         Younger than 30 Younger than 30 40 - 50
```

```
## [129] 40 - 50         30 - 40         Younger than 30 30 - 40
## [133] 30 - 40         Younger than 30 Younger than 30 30 - 40
## [137] 30 - 40         40 - 50         Younger than 30 30 - 40
## [141] 40 - 50         Younger than 30 30 - 40         40 - 50
## [145] 40 - 50         Younger than 30 Younger than 30
## Levels: Younger than 30 30 - 40 40 - 50 Older than 50
```

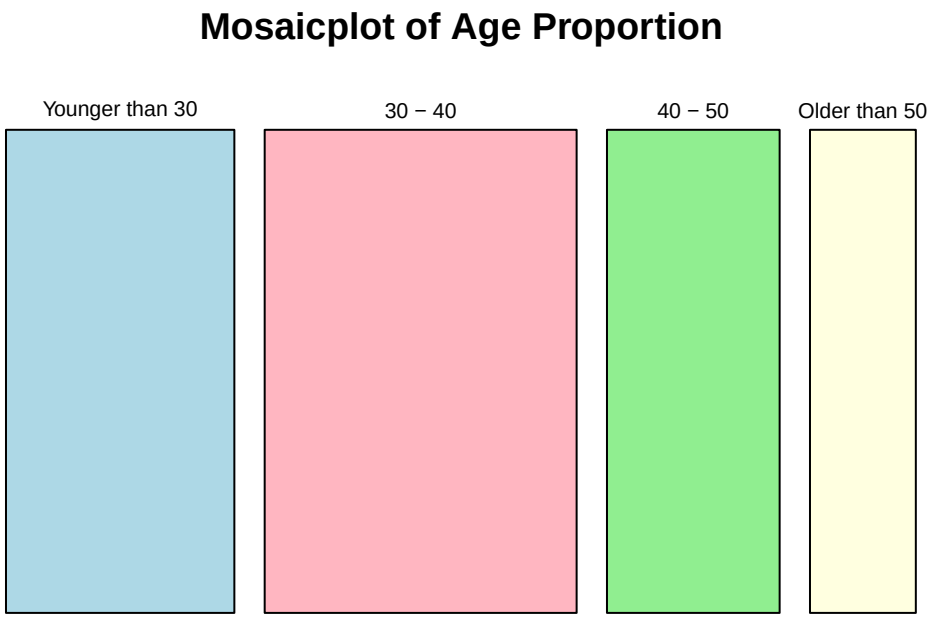
(b)

```
table(cate_age)
```

```
## cate_age
## Younger than 30         30 - 40         40 - 50   Older than 50
##              41              56              31              19
```

(c)

```
p_table <- table(cate_age)/length(data.twins$AGE)
mosaicplot(p_table, color = c("lightblue","lightpink","lightgreen", "lightyellow"),
main = "Mosaicplot of Age Proportion", xlab = "")
```



Question 4:

(a)

```r
cate_salary <- cut(data.twins$HRWAGEL, breaks = c(0, 7, 13, 20, 150), labels = c(
"Less than 7", "7 - 13", "13 - 20","20 - 150") )
print(cate_salary)
```

```
##   [1] 7 - 13      7 - 13      13 - 20     13 - 20     7 - 13      13 - 20
##   [7] 7 - 13      13 - 20     7 - 13      20 - 150    Less than 7 7 - 13
##  [13] 7 - 13      13 - 20     13 - 20     7 - 13      13 - 20     7 - 13
##  [19] Less than 7 20 - 150    Less than 7 Less than 7 7 - 13      7 - 13
##  [25] 20 - 150    20 - 150    Less than 7 7 - 13      13 - 20     13 - 20
##  [31] Less than 7 13 - 20     20 - 150    Less than 7 20 - 150    20 - 150
##  [37] 7 - 13      7 - 13      Less than 7 20 - 150    7 - 13      7 - 13
##  [43] 7 - 13      13 - 20     20 - 150    7 - 13      Less than 7 Less than 7
##  [49] 7 - 13      13 - 20     7 - 13      13 - 20     13 - 20     Less than 7
##  [55] 7 - 13      Less than 7 20 - 150    Less than 7 7 - 13      Less than 7
##  [61] 13 - 20     20 - 150    Less than 7 13 - 20     Less than 7 Less than 7
##  [67] Less than 7 13 - 20     7 - 13      7 - 13      7 - 13      7 - 13
##  [73] 13 - 20     7 - 13      7 - 13      7 - 13      13 - 20     Less than 7
##  [79] 7 - 13      7 - 13      20 - 150    7 - 13      7 - 13      13 - 20
##  [85] 7 - 13      20 - 150    7 - 13      7 - 13      Less than 7 Less than 7
##  [91] Less than 7 13 - 20     Less than 7 20 - 150    13 - 20     Less than 7
##  [97] Less than 7 20 - 150    7 - 13      7 - 13      7 - 13      7 - 13
## [103] 13 - 20     7 - 13      7 - 13      Less than 7 7 - 13      7 - 13
## [109] 20 - 150    13 - 20     7 - 13      Less than 7 7 - 13      7 - 13
## [115] Less than 7 Less than 7 7 - 13      Less than 7 Less than 7 7 - 13
## [121] 13 - 20     7 - 13      Less than 7 13 - 20     Less than 7 7 - 13
## [127] Less than 7 13 - 20     Less than 7 7 - 13      13 - 20     13 - 20
## [133] 13 - 20     Less than 7 7 - 13      7 - 13      13 - 20     Less than 7
## [139] Less than 7 7 - 13      13 - 20     Less than 7 20 - 150    13 - 20
## [145] Less than 7 13 - 20     13 - 20
## Levels: Less than 7 7 - 13 13 - 20 20 - 150
```

```r
table(cate_salary)
```

```
## cate_salary
## Less than 7      7 - 13     13 - 20    20 - 150
##          40          55          35          17
```

(b)

```r
table(cate_age, cate_salary)
```

```
##                  cate_salary
## cate_age          Less than 7 7 - 13 13 - 20 20 - 150
##   Younger than 30          18     16       7        0
##   30 - 40                  12     25      15        4
##   40 - 50                   7      7       7       10
##   Older than 50             3      7       6        3
```

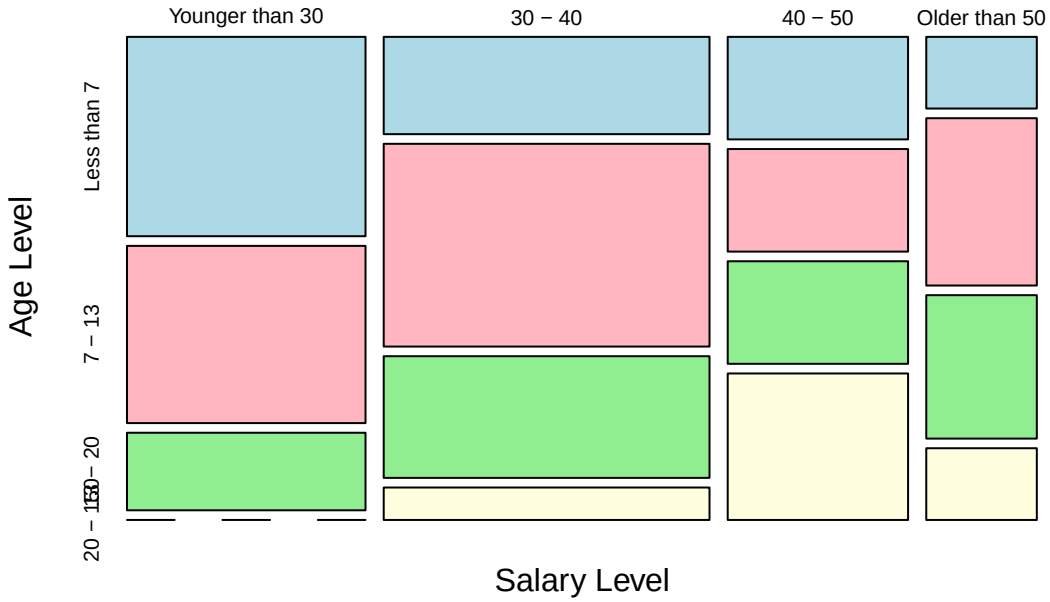
(c)

```
age_salary <- prop.table(table(cate_age, cate_salary))
print(age_salary)
```

```
##                  cate_salary
## cate_age          Less than 7     7 - 13    13 - 20   20 - 150
##   Younger than 30  0.12244898 0.10884354 0.04761905 0.00000000
##   30 - 40          0.08163265 0.17006803 0.10204082 0.02721088
##   40 - 50          0.04761905 0.04761905 0.04761905 0.06802721
##   Older than 50    0.02040816 0.04761905 0.04081633 0.02040816
```

(d)

```
mosaicplot(age_salary, color = c("lightblue", "lightpink", "lightgreen", "lightyellow"),
xlab = "Salary Level", ylab = "Age Level", main = "")
```



(e) Young people are less possible to have an extremely high salary. For those people who has extremely high salary, most of them are people older than 40 years old but younger than 50 years old.

Question 5:

(a)

```r
chisq.test(table(cate_age, cate_salary))
```

```
## Warning in chisq.test(table(cate_age, cate_salary)): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(cate_age, cate_salary)
## X-squared = 27.628, df = 9, p-value = 0.0011
```

From the P-value above, we can see that the null hypothesis is rejected, and the colums are not independent from rows.

(b)

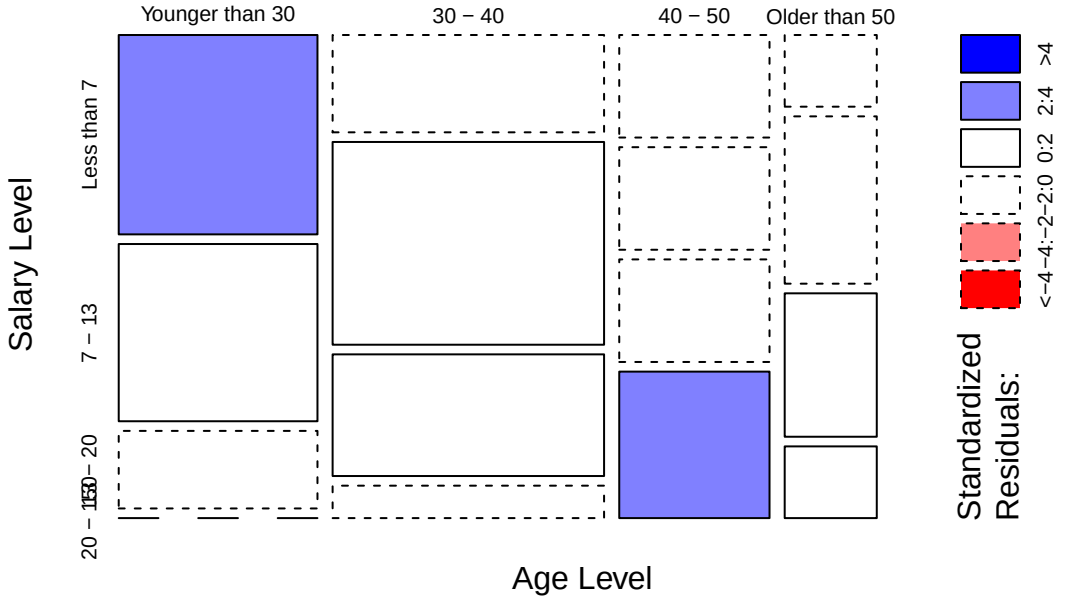
```r
X <- table(cate_age, cate_salary)
A <- matrix(rep(rowSums(X),4),4,4, byrow = T)
B <- matrix(rep(colSums(X),4),4,4)
tot <- sum(X)
Expected <- t(A*B/tot)
Pearson <- (X-Expected)/sqrt(Expected)
print(Pearson)
```

```
##                  cate_salary
## cate_age          Less than 7      7 - 13     13 - 20    20 - 150
##   Younger than 30  2.04888409  0.16847668 -0.88397792 -2.17749778
##   30 - 40         -0.82951506  0.88426603  0.45643546 -0.97302555
##   40 - 50         -0.49421161 -1.35028622 -0.14022146  3.38803367
##   Older than 50   -0.95438855 -0.04082284  0.69404918  0.54152945
```

So, the people younger than 30 with salary less than 7 or more than 20 and the people with age between 40-50 with salary 20-150 has Pearson's residual absolutely larger than 2.

(c)

```r
mosaicplot(table(cate_age, cate_salary), shade = TRUE, main = "",
xlab = "Age Level", ylab = "Salary Level")
```



If the columns and rows are independent, it should satisfy the following equation:

$$P(X = x,\ Y = y) = P(X = x) \times P(Y = y)$$

And the frequency of each group should also follows that distribution. However, after comparing the expected value and the true value, We find the people who is younger than 30 has a extremely higher frequency to get salary less than 7 and extremely lower frequency to get a high salary. Furthermore, the frequency observed is extremely higher than expected for people from 40-50 to get a salary 20-150.

Question 6:

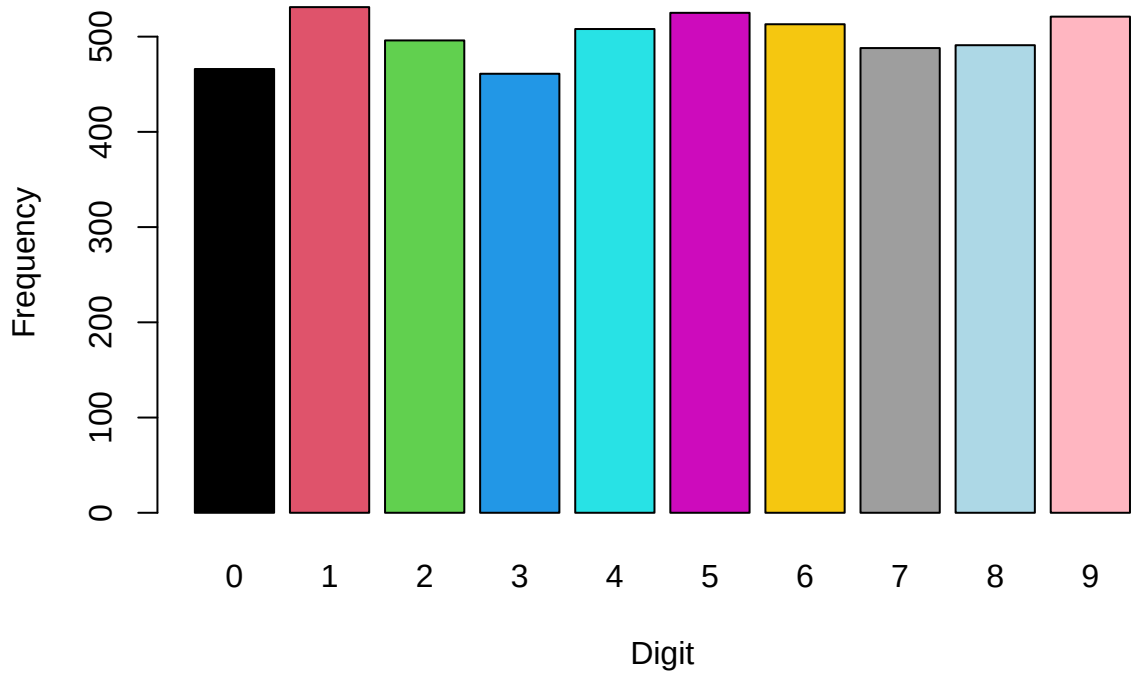
(a)

```
pidigits =
read.table("http://www.itl.nist.gov/div898/strd/univ/data/PiDigits.dat",
skip=60)
table(pidigits)
```

```
## pidigits
##   0   1   2   3   4   5   6   7   8   9
## 466 531 496 461 508 525 513 488 491 521
```

(b)

```
barplot(table(pidigits), col = c(1:8, "lightblue", "lightpink"),
xlab = "Digit", ylab = "Frequency")
```



(c)

```
chisq.test(table(pidigits))
```

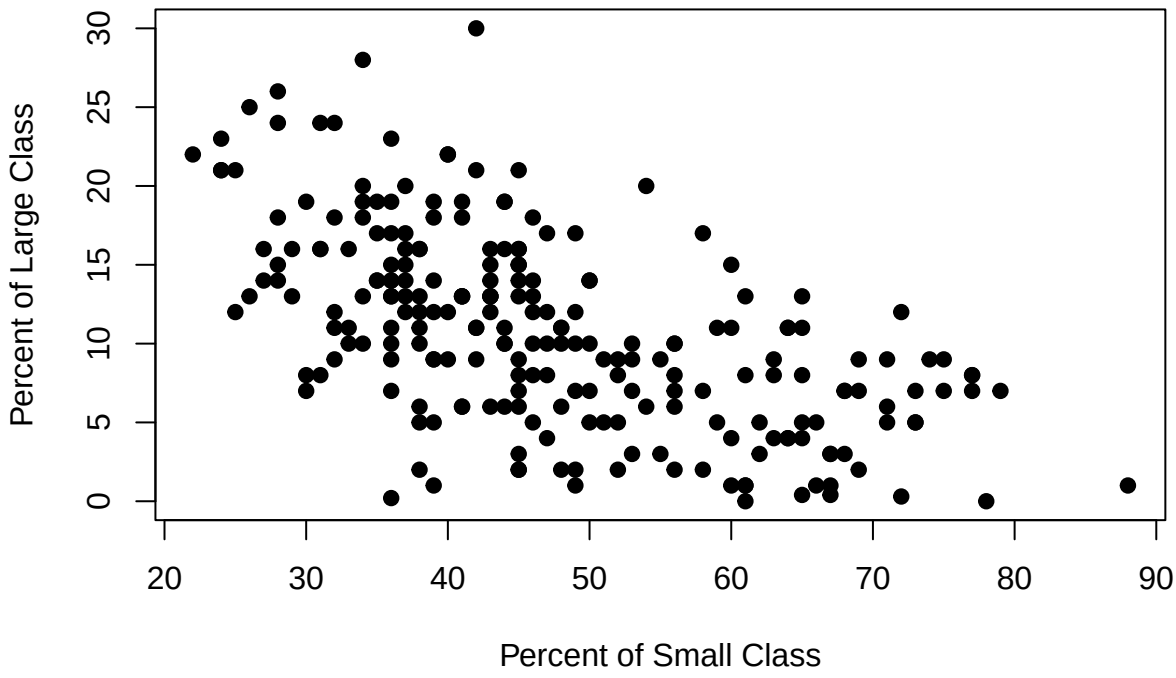
```
##
##  Chi-squared test for given probabilities
##
## data:  table(pidigits)
## X-squared = 10.356, df = 9, p-value = 0.3224
```

From the chi-squared test above, we cannot reject the null hypothesis that the digits are randomly distributed from 0 to 9.

Question 7:

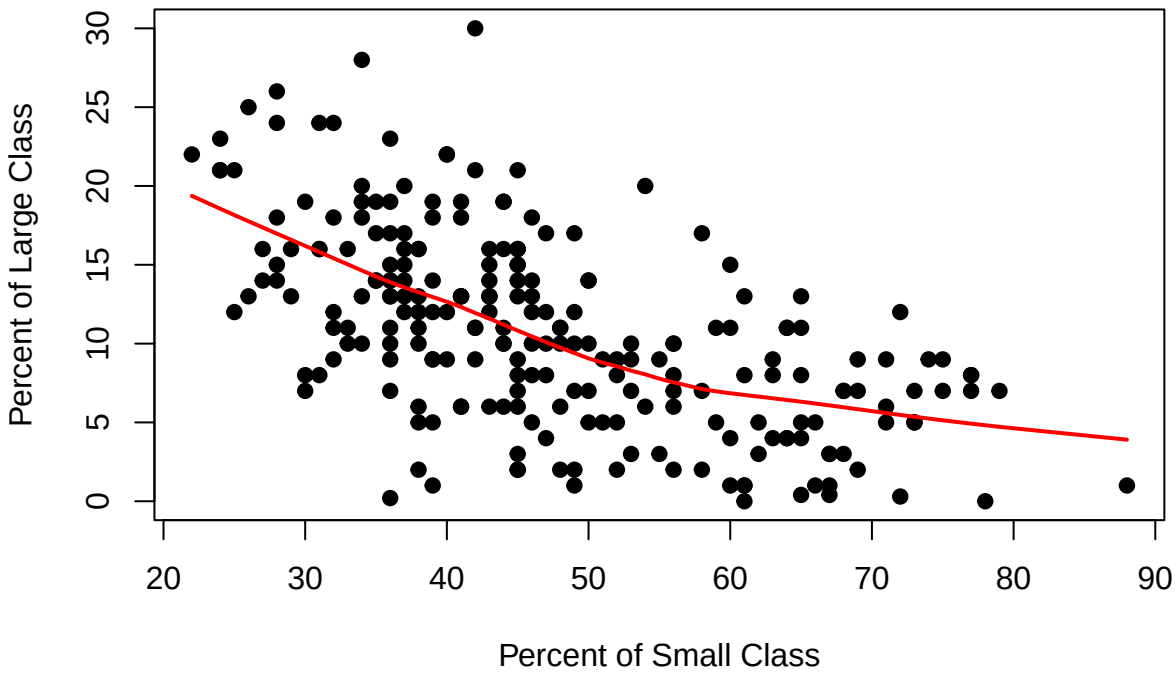
(a)

```
college <- read.csv("D:/77/UCSC/study/204/HW/college.csv", header = T)
college <- na.omit(college)
attach(college)
plot(Pct.20, Pct.50, pch = 19, xlab = "Percent of Small Class",
ylab = "Percent of Large Class")
```



(b)

```
plot(Pct.20, Pct.50, pch = 19, xlab = "Percent of Small Class",
ylab = "Percent of Large Class")
lines(lowess(Pct.20, Pct.50), lwd = 2, col = "red")
```



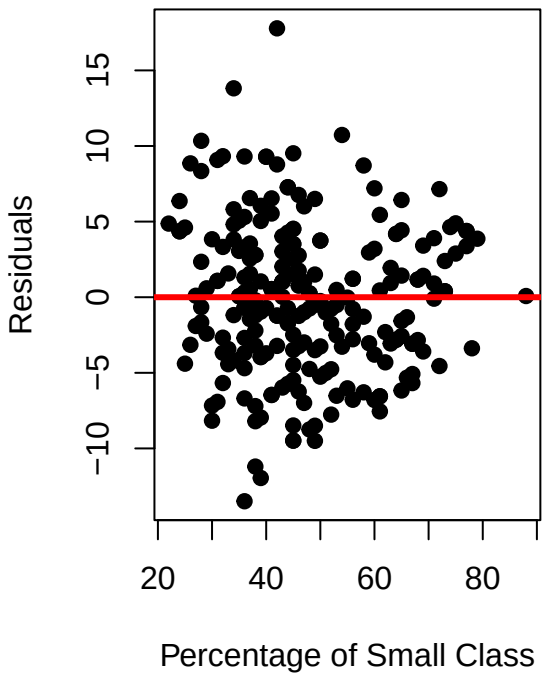
(c)

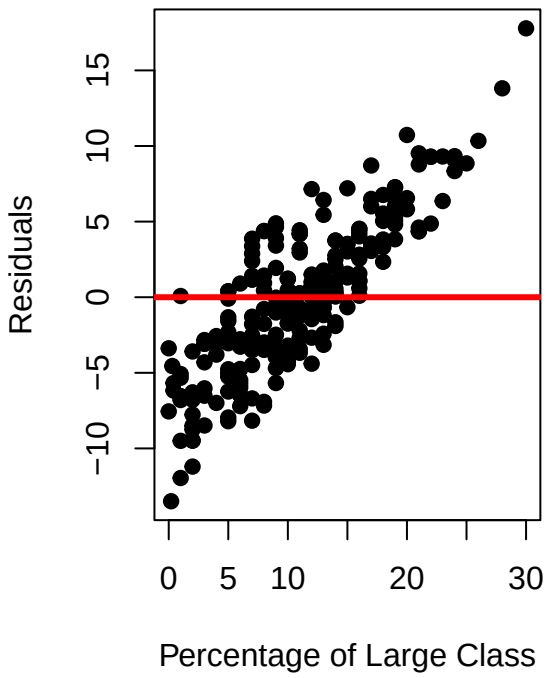
From the red line above we can see the percent of large class in this school is around 9%.

(d)

```
par(mfrow = c(1, 2))
fit <- line(x = Pct.20, y = Pct.50)
plot(Pct.20, fit$residuals, pch = 19, xlab = "Percentage of Small Class",
ylab = "Residuals")
abline(h = 0, lwd = 3, col = "red")


plot(Pct.50, fit$residuals, pch = 19, xlab = "Percentage of Large Class",
ylab = "Residuals")
abline(h = 0, lwd = 3, col = "red")
```





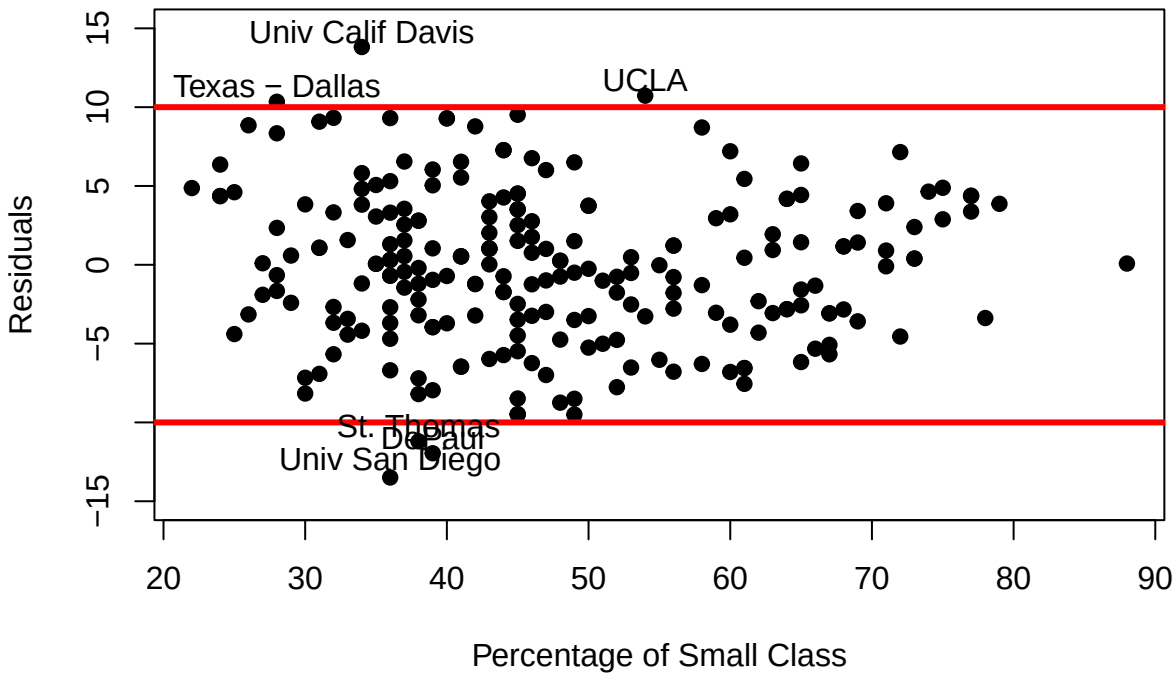
I think as the percentage of large classes increase, the residuals seem to have a increasing trend. However, for the residual plot between small classes and residuals, it seems the variance of residuals are becoming smaller as the percentage of small classes increases.

(e)

As the percentage of small classes increases, the residuals seems to become more concentrated around 0. It could be a specific pattern of residuals which the residuals decrease as the percentage of small classes increase.

(f)

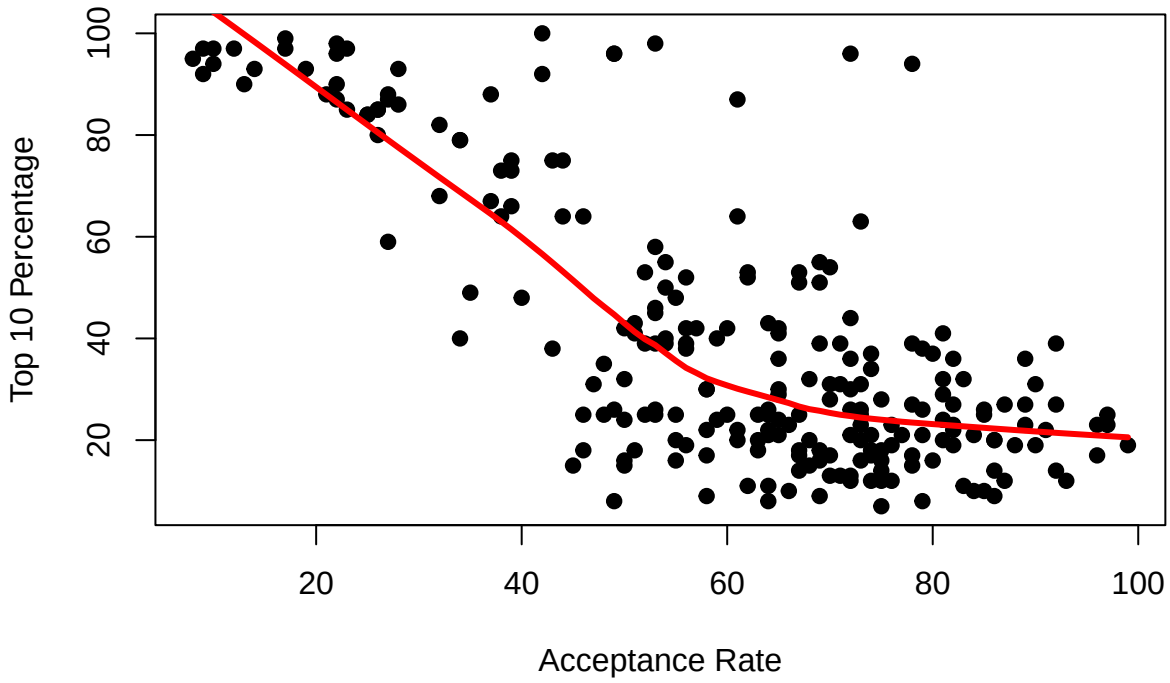
```
plot(Pct.20, fit$residuals, pch = 19, xlab = "Percentage of Small Class",
ylab = "Residuals", ylim = c(-15, 15))
abline(h = 10, col = "red", lwd = 3)
abline(h = -10, col = "red", lwd = 3)
#identify(Pct.20, fit$residuals, n = 6, labels = School)
#After selecting the points, the points are as follows: 24 43 112 138 148 185 186
outlier <- c(24, 43, 112, 138, 185, 186)
text(x = Pct.20[outlier], y = fit$residuals[outlier]+1, labels = School[outlier] )
```



Question 8:

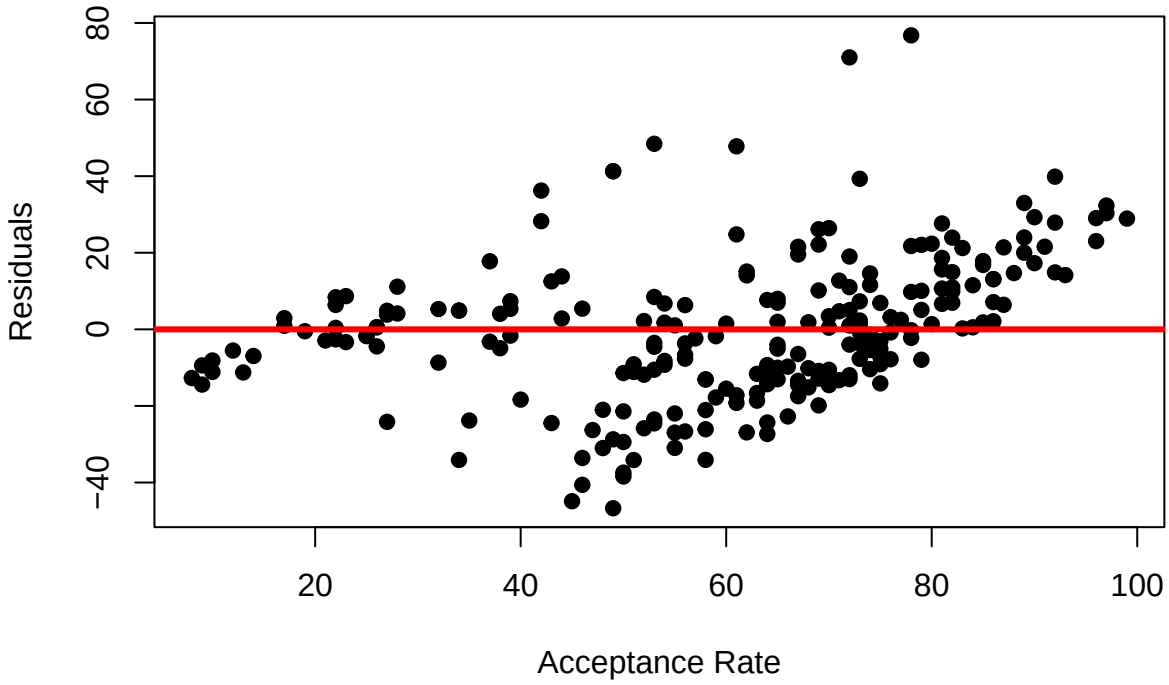
(a)

```
plot(Accept.rate, Top.10, pch = 19, xlab = "Acceptance Rate", ylab = "Top 10 Percentage")
lines(lowess(Accept.rate, Top.10), col = "red", lwd = 3)
```



From the chart above, we can see there is a negative correlation between the acceptance rate and the top 10 percentage. Then, we will try some linear model.

```
fit <- line(Accept.rate, Top.10)
plot(Accept.rate, fit$residuals, pch = 19, xlab = "Acceptance Rate", ylab = "Residuals")
abline(h = 0, lwd = 3, col = "red")
```



From the residual plot, we can see the residuals are not normally distributed, we should consider another model, it seems that the residuals increases first and decreases, and then increases again, so we may need a quadratic function to fit for this model.

(b)

From the scatter plot in the question (a), we can see the plot is more concentrated distributed in two areas where acceptance rate is either small or big. Also, from the lowness function above, it showed that the slope changes differently from one cluster to the other. So I think the schools are always divided into elite and non-elite schools.

Question 9:

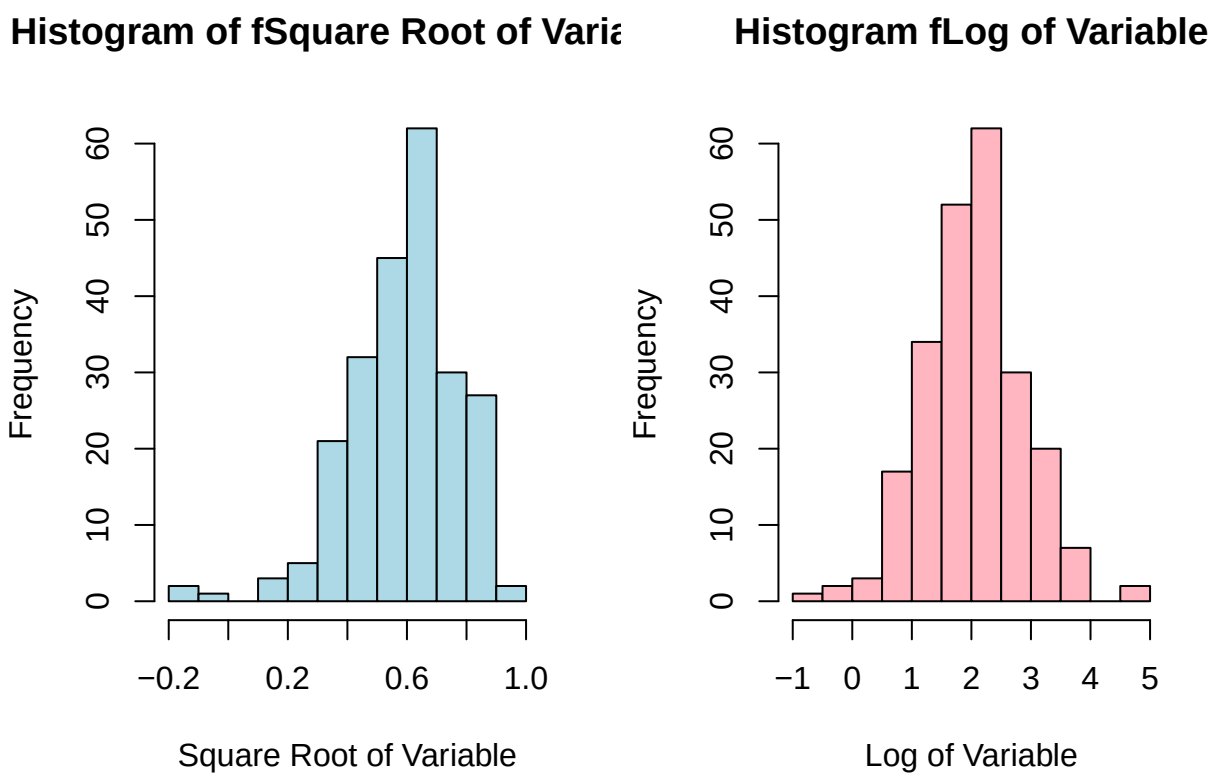
(a)

```
hist(Full.time, col = "lightblue", xlab = "Faculty Hired Full-time",
ylab = "Frequency", main = "")
```

This is a left-skewed distribution, with a larger percentage of faculties are hired full time.

(b)

```
froot_full <- sqrt(Full.time/100) - sqrt(1-Full.time/100)
flog_full <- log(Full.time/100 + 0.01) - log(1-Full.time/100 + 0.01)
par(mfrow = c(1,2))
hist(froot_full, main = "Histogram of fSquare Root of Variable",
xlab = "Square Root of Variable", ylab = "Frequency", col = "lightblue")
hist(flog_full, main = "Histogram fLog of Variable",
xlab = "Log of Variable", ylab = "Frequency", col = "lightpink")
```



I think flog transformation is better than the fsquare root transformation.

(c)

I will choose the fLog transformation since it is more likely to be a normal.

```
mu <- mean(flog_full)
sigma <- var(flog_full)
sd <- sqrt(sigma)
lower <- mu - sd
upper <- mu + sd
matrix(c(round(lower,2), round(upper,2)),1,2)
```

```
##      [,1] [,2]
## [1,] 1.21 2.91
```

So this is the interval that includes around 68% of data in the data set.

PS: Here there includes some value that are infinity, I simply changed the flog function a little. I did not omit the data since all data points reveal information of the population. Also I can use the quantile function:

```
quantile(flog_full, c(0.16, 0.8))
```

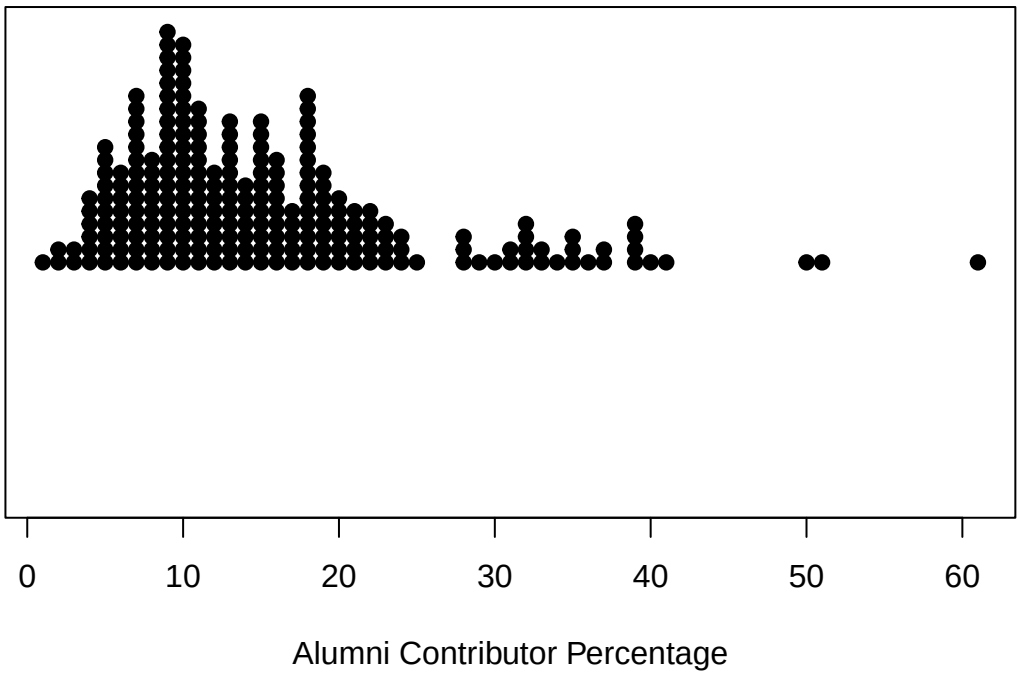
```
##      16%      80%
## 1.233954 2.772589
```

They are similar to each other.

Question 10:

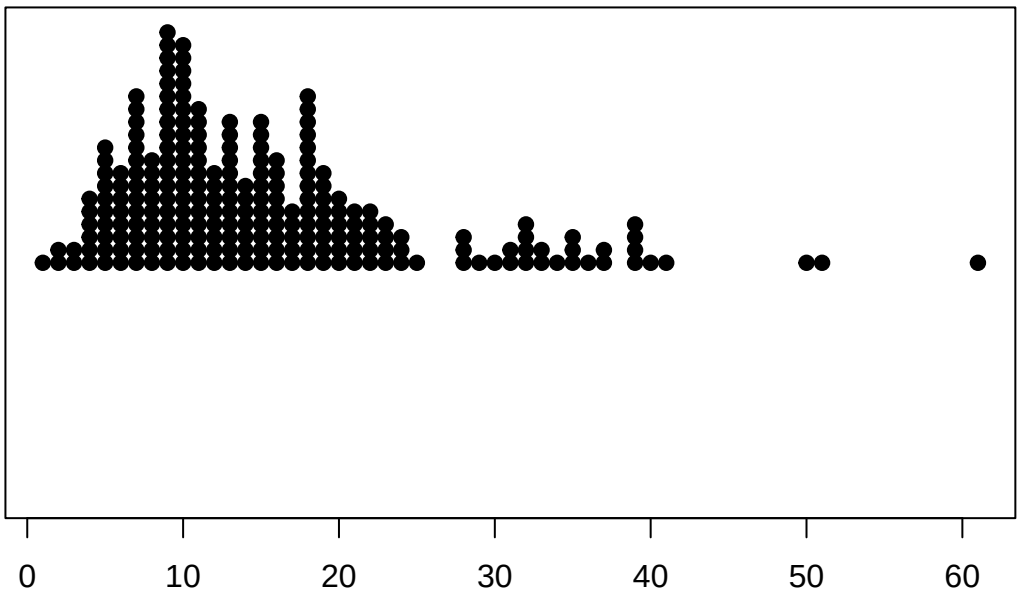
(a)

```
stripchart(Alumni.giving, method = "stack", pch = 19, xlab = "Alumni Contributor Percentage")
```



(b)

```
stripchart(Alumni.giving, method = "stack", pch = 19, xlab = "Alumni Contributor Percentage")
```

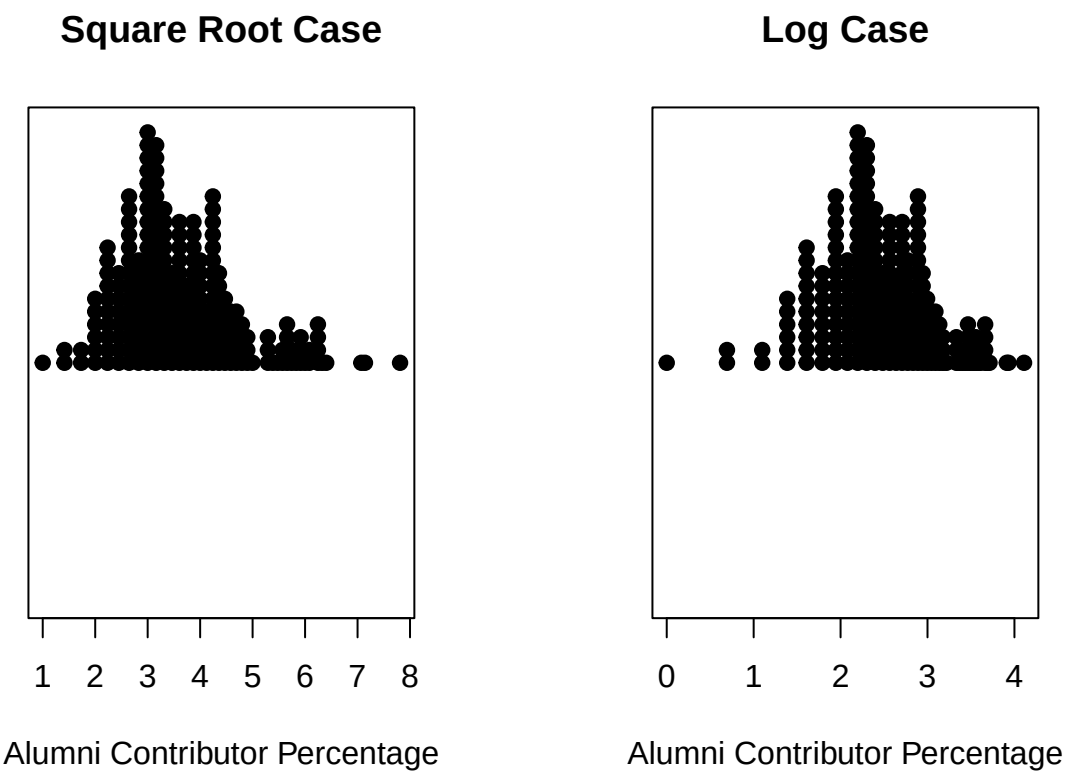



```
#identify(Alumni.giving, n = 3, labels = School, tolerance = 1)
School[which(Alumni.giving > 45)]
```

```
## [1] "Princeton"  "Dartmouth"  "Notre Dame"
```

(c)

```
roots_alu = sqrt(Alumni.giving)
logs_alu = log(Alumni.giving)
par(mfrow = c(1,2))
stripchart(roots_alu, method = "stack", pch = 19, xlab = "Alumni Contributor Percentage",
main = "Square Root Case")
stripchart(logs_alu, method = "stack", pch = 19, xlab = "Alumni Contributor Percentage",
main = "Log Case")
```

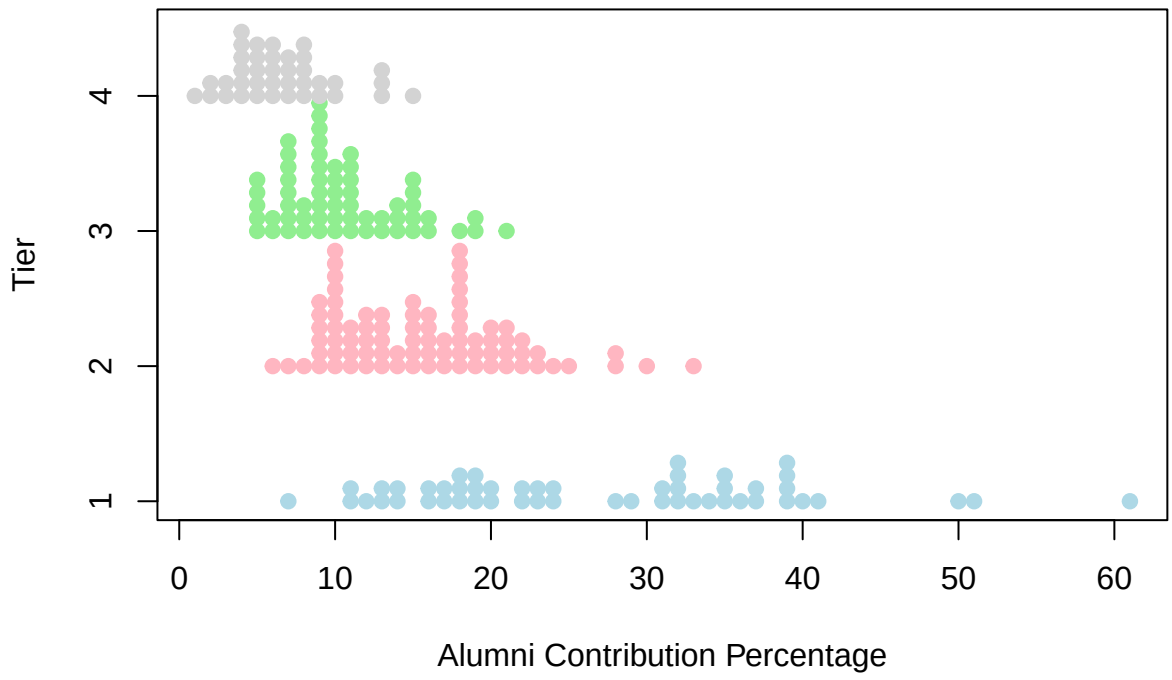


From the Chart above, we can see that the log transformation is better since it looks more likely to be an approximately normal distribution.

Question 11:

(a)

```
stripchart(Alumni.giving~Tier, method = "stack", pch = 19,
col = c("lightblue", "lightpink", "lightgreen", "lightgray"),
xlab = "Alumni Contribution Percentage", ylab = "Tier")
```

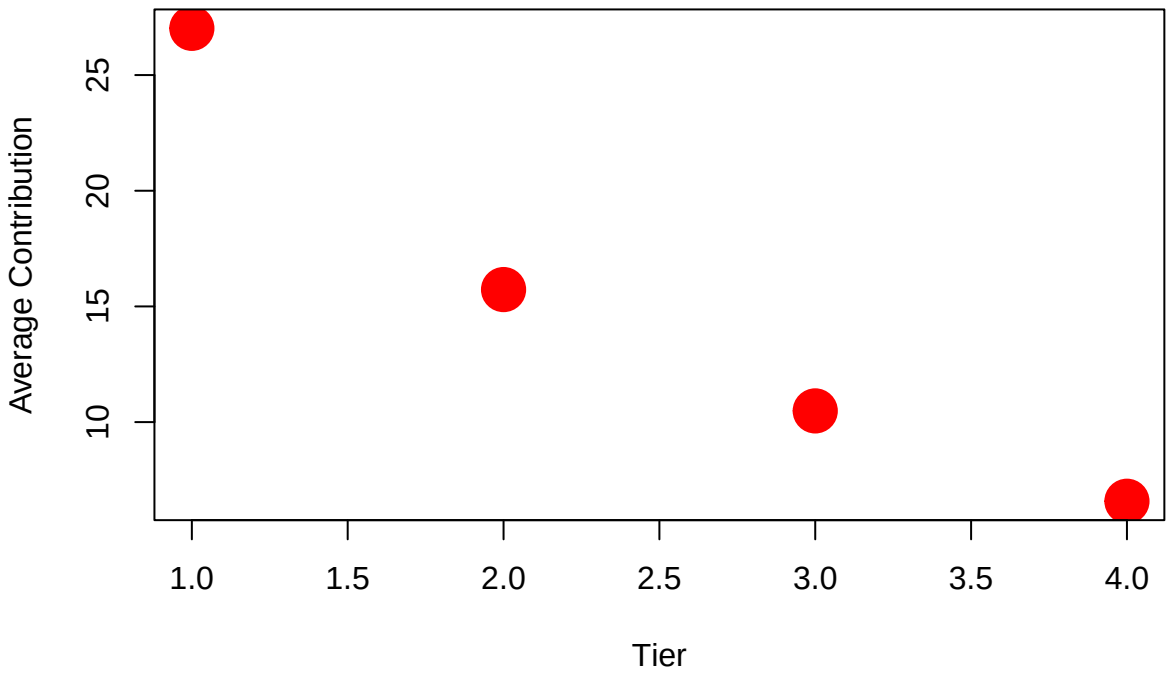


(b)

```
mu1 <- mean(Alumni.giving[which(Tier==1)])
mu2 <- mean(Alumni.giving[which(Tier==2)])
mu3 <- mean(Alumni.giving[which(Tier==3)])
mu4 <- mean(Alumni.giving[which(Tier==4)])
RES <- as.matrix(c(mu1, mu2, mu3,mu4))
rownames(RES) <- c("Tier 1","Tier 2","Tier 3","Tier 4")
print(RES)
```

```
##             [,1]
## Tier 1 27.019608
## Tier 2 15.728395
## Tier 3 10.483333
## Tier 4  6.578947
```

```
plot(RES, xlab = "Tier", ylab = "Average Contribution", pch = 19, cex = 3, col = "red")
```



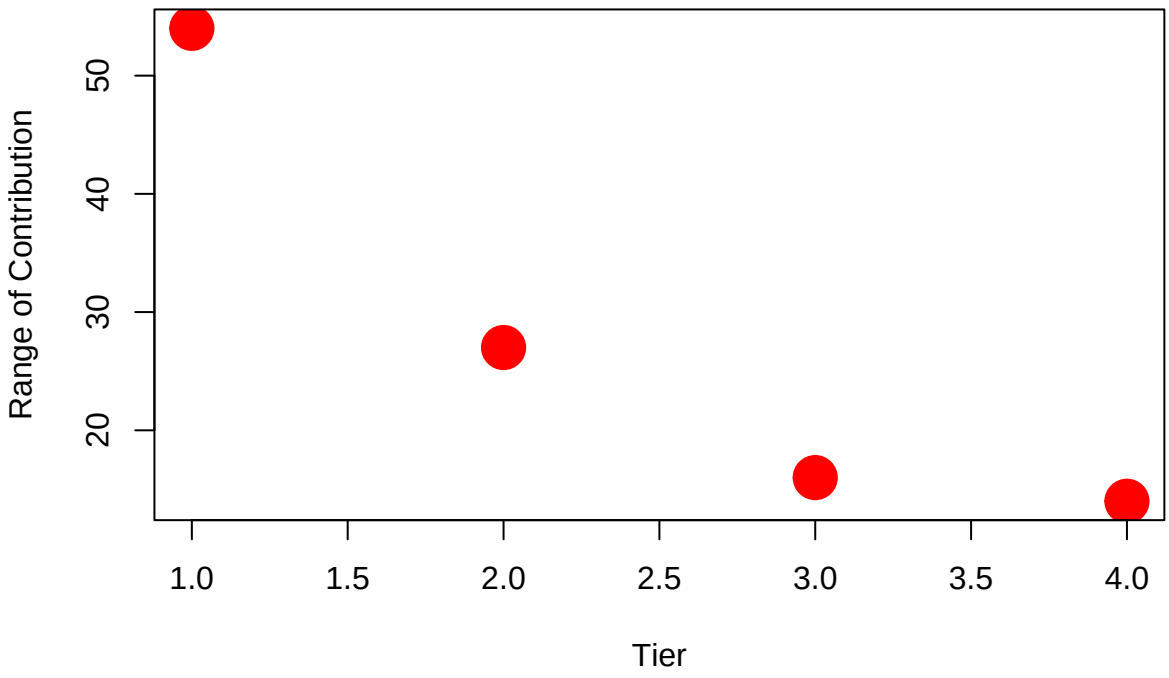
From the picture above, we can see that school from tier one has the most alumni contribution, tier 2 follows it, then tier 3, at last tier 4. Better school have more alumni contributions. As one moves from 4 to 1, the average is increasing.

(c)

```
range1 <- diff(range(Alumni.giving[which(Tier==1)]))
range2 <- diff(range(Alumni.giving[which(Tier==2)]))
range3 <- diff(range(Alumni.giving[which(Tier==3)]))
range4 <- diff(range(Alumni.giving[which(Tier==4)]))
RAN <- as.matrix(c(range1, range2, range3, range4))
rownames(RAN) <- c("Tier 1","Tier 2","Tier 3","Tier 4")
print(RAN)
```

```
##        [,1]
## Tier 1   54
## Tier 2   27
## Tier 3   16
## Tier 4   14
```

```
plot(RAN, xlab = "Tier", ylab = "Range of Contribution", pch = 19, cex = 3, col = "red")
```

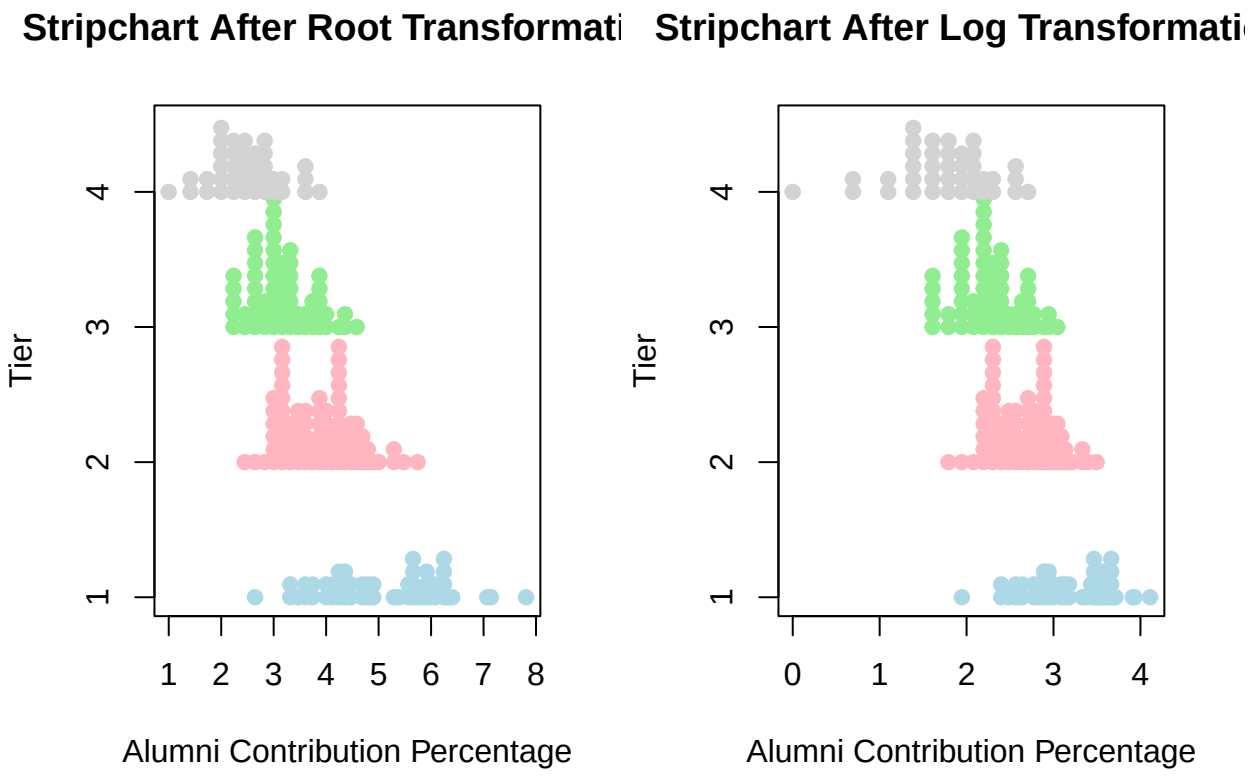


From the plot, we can see that the Tier one has the largest range. Then as the tier goes down, the range becomes smaller. As one moves from 4 to 1, the range is increasing.

(d)

```
par(mfrow = c(1,2))
stripchart(roots_alu~Tier, method = "stack", pch = 19,
col = c("lightblue", "lightpink", "lightgreen", "lightgray"),
xlab = "Alumni Contribution Percentage", ylab = "Tier",
main = "Stripchart After Root Transformation")


stripchart(logs_alu~Tier, method = "stack", pch = 19,
col = c("lightblue", "lightpink", "lightgreen", "lightgray"),
xlab = "Alumni Contribution Percentage", ylab = "Tier",
main = "Stripchart After Log Transformation")
```



(e)

Root transformation successfully makes the spread almost the same for only tier 2, 3 and 4, but the spread of tier 1 is still obviously larger than the other three. For the log transformation, it makes the spread of tier 1, 2 and 3 almost the same but make tier 4 a little more widely spreaded than the other three.