Homework 1

Thu Nguyen

14 April, 2019

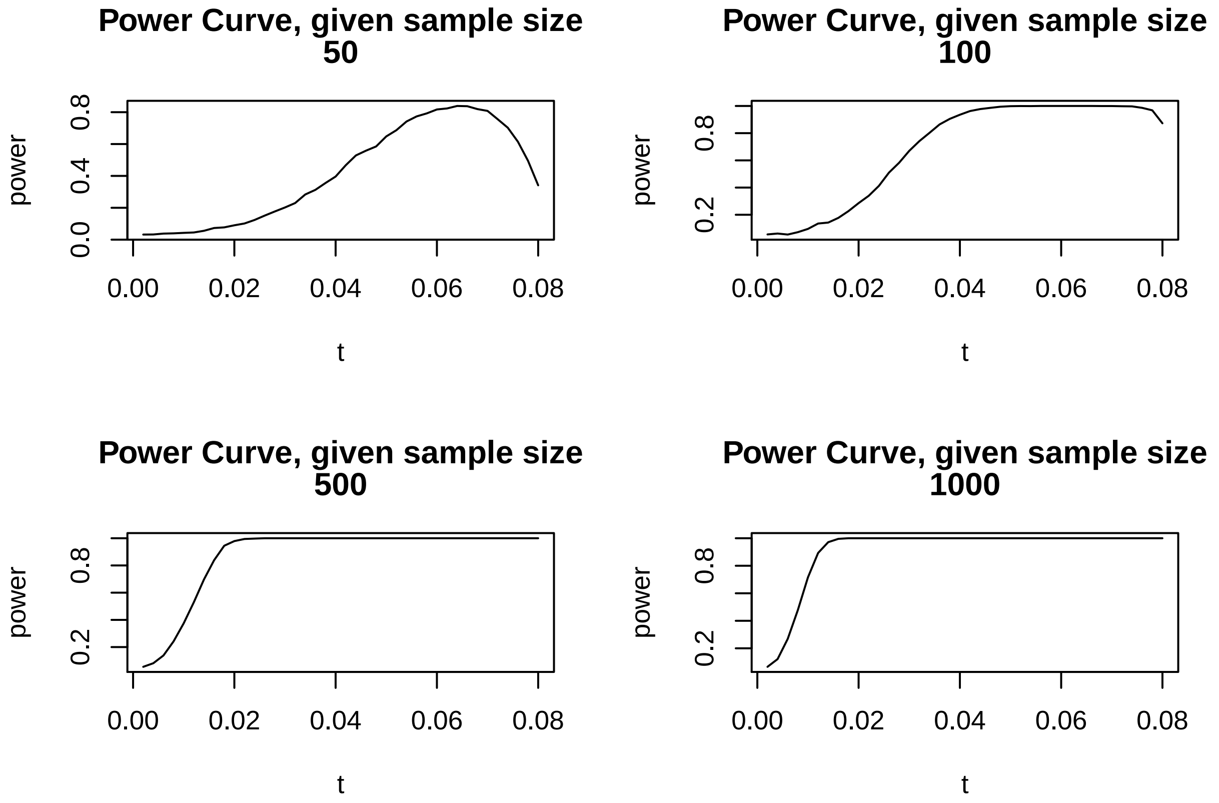
Problem 1

Part A

```r
# Problem 1  ---------------------------------------------------
### Part A   ---------------------------------------------------
chisq.power = function( k, t, n, B = 2000) {

  # k: parameter for probability under H0: Unif(1/2k)
  # t: parameter for probability under H1
  # n: number of data points
  # B: number of simulations

  # binary vector R of length B
  R <- numeric(B)

  # Probability vector under Alternative Hypothesis, H1
  h1_probvector <- numeric(2*k)
  # if j <= k
  for (j in 1:k) {
    h1_probvector[j] <- 1/(2*k) + t
  }
  # if j > k
  for (j in (k+1):(2*k)) {
    h1_probvector[j] <- 1/(2*k) - t
  }

  # Monte Carlo simulation
  for (b in 1:B) {
    # samples from MC simulation generated based on H1 pdf of size 2k
    MCsamples <- sample( 1:(2*k), n, replace = TRUE, prob = h1_probvector)
    # convert MC simulatin to frequency table
    MCsamples <- table(MCsamples)
    # Chi-squared test of the MCsamples: if p-val <= .05, R[b] = 1
    if (chisq.test(MCsamples)$p.value <= .05) {
      R[b] = 1
    } else {
      R[b] = 0
    }
  }

  # Proportion of Correctly Rejecting H0 given H1
  return(sum(R)/B)
}
```

Part B

```r
### Part B   ----------------------------------------------------
# vector of different sample size n
n <- c(50, 100, 500, 1000)
# k = 6, as given in question
k <- 6
# sequence vector of t: 0 < t < 1/2k
t <- seq(0+.002, 1/(2*k) - .002, by = .002)

# null vector of powers of length(t): power = f(t)
powers <- numeric(length(t))

# layout plots 2x2
par(mfrow = c(2,2))

# loop over different sample size n
for (j in 1:length(n)) {
  # loop over different t to find corresponding power of that t
  for (i in 1:(length(t))) {
    powers[i] <- chisq.power(k, t[i], n[j])
    i <- i + 1
  }
  # plots
  plot(t, powers, type = 'l', xlab = 't', ylab = 'power',
       main = append('Power Curve, given sample size', n[j]))
}
```



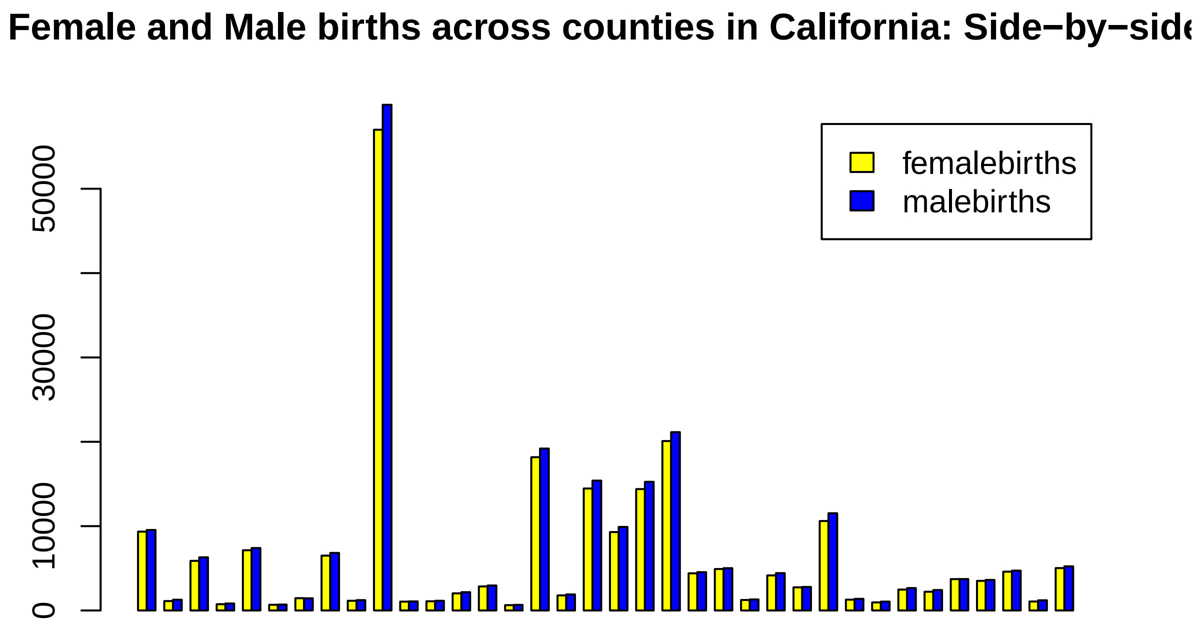
Problem 2

Data pre-processing

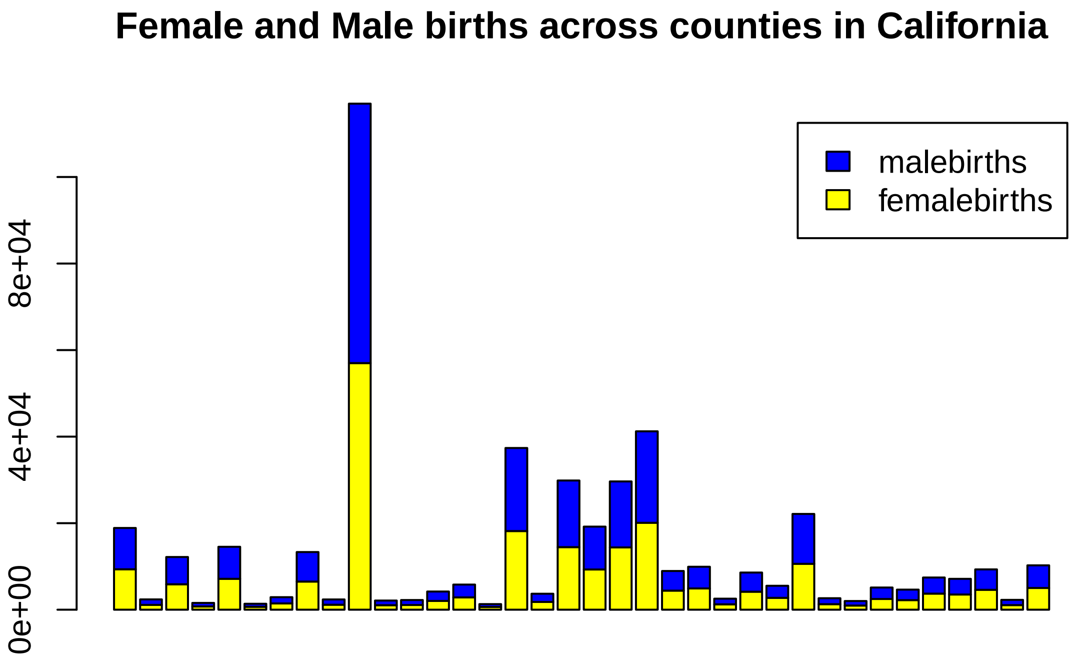
```
# Problem 2  ----------------------------------------------------
# Preprocess data
mydata <- read.table('natality-california-2017.txt', header = TRUE)
save(mydata, file = 'natality-california-2017.rda')
load('natality-california-2017.rda')

# Vector of Female births
femalebirths <- mydata[1:36,5]
# Vector of Male births
malebirths <- mydata[37:72,5]
# Joint matrix of Female & Male births
jointbirths <- rbind(femalebirths, malebirths)
```

Some exploratory plots

```
# Barplot of Female vs. Male births per County
barplot(jointbirths, col = c('yellow', 'blue'), beside = T, legend = T,
        main = 'Female and Male births across counties in California: Side-by-side')
```



```r
barplot(jointbirths, col = c('yellow', 'blue'), legend = T,
        main = 'Female and Male births across counties in California')
```



Comments

From the barplot above, there does not seem to be considerable differences between the number of births between females and males across California counties.

Hypothesis Test:

Let p_i be the probability of a girl being born in county i, where i is 1 of the 36 counties in California.

$$H_0 : (p_1, p_2, \ldots, p_{36}) = (p_1^0, p_2^0, \ldots, p_{36}^0) : p_1^0 = p_2^0 = \cdots = p_{36}^0$$

$$H_1 : (p_1, p_2, \ldots, p_{36}) \neq (p_1^0, p_2^0, \ldots, p_{36}^0) : p_1^0 = p_2^0 = \cdots = p_{36}^0$$

Chi-squared test:

```r
# Chi-squared test
chisq.test(jointbirths)
```

```

	Pearson's Chi-squared test

data:  jointbirths
X-squared = 41.285, df = 35, p-value = 0.215
```

Comments

Given the p value of .215, for any level of significance less than 21.5%, we would fail to reject H_0: The chances of a baby being born a girl are the same across counties in California.

Problem 3

```r
chisq.perm.test = function(tab, B = 2000) {

  # Obeserved statistic from tab: as reference
  Dobserved <- chisq.test(tab)$stat

  # Dimensions of tab
  nrows <- nrow(tab)
  ncols <- ncol(tab)


  # Index each categorical element in rows and columns
  # Vector of numbers in each row
  # null rows
  rows <- numeric(nrows)
  totalrows <- numeric(0)
  # vector of 1s, 2s, 3s, 4s, indicating row number of original tab
  for (i in 1:nrows) {
    rows[i] <- sum(tab[i,])
    totalrows <- c(totalrows, rep.int(i, times = rows[i]))
  }
  # Vector of numbers in each column
  columns <- numeric(ncols)
  totalcols <- numeric(0)
  # vector of 1s, 2s, 3s, 4s, indicating column number of original tab
  for (i in 1:ncols) {
    columns[i] <- sum(tab[,i])
    totalcols <- c(totalcols, rep.int(i, times = columns[i]))
  }

  # counter of #(stat > Doberved)
  count <- 0

  # vector of permutations' Chi-sq. test-stat
  D <- numeric(0)

  # Total counts of tab
  total <- sum(tab[,])

  ###############################################
  for (b in 1:B) {
    # fix x-values
    x <- totalrows
    # randomly shuffle y-values
    y <- sample(totalcols, total, replace = FALSE)
    # new data.frame of the shuffling/permutation
    matrixperm <- cbind(x, y)

    # matrixpermcount: count the pairs after permutations
    # null matrixpermcount of only zeros
    matrixpermcount <- matrix(0, nrow = nrows, ncol = ncols)

    # After permutation: Rearrange into matrixpermcount
    # index of x for each row: row 1-4
    for (x in 1:nrows) {
      # index of y for each column: column 1-4
      for (y in 1:ncols) {
        # index of m for looping through each row in matrixperm
        for (m in 1:total) {
          # Comparing & sorting the indexed cells (1,1), (2,2), ...
          # Once located the right cells, count #(occurances)
          if (all(matrixperm[m,] == c(x, y)) == TRUE) {
            # all TRUE <=> cell located
            # increase count in the cell in new location
            matrixpermcount[x, y] <- matrixpermcount[x, y] + 1
          }
        }
      }
    }

    # Chi-sq test-statistic for that permutation
    D[b] <- chisq.test(matrixpermcount)$stat

    # Compare Chi-sq test-statistic from the permutation
    # with Oberserved test-statistic
    # if D_perm > D_obs: increase count
    if (D[b] >= Dobserved) {
      count = count + 1
    }
  }
  # Proportion of permutation returning test-statistic more extreme
  # than D_obs
  return((count+1)/(B+1))
}
```

Testing on the `HairEyeColor` dataset

```
tab <- apply(HairEyeColor, c(1,2), sum)
tab
```

```
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
```

```
chisq.perm.test(tab)
```

```
[1] 0.0004997501
```

Problem 4

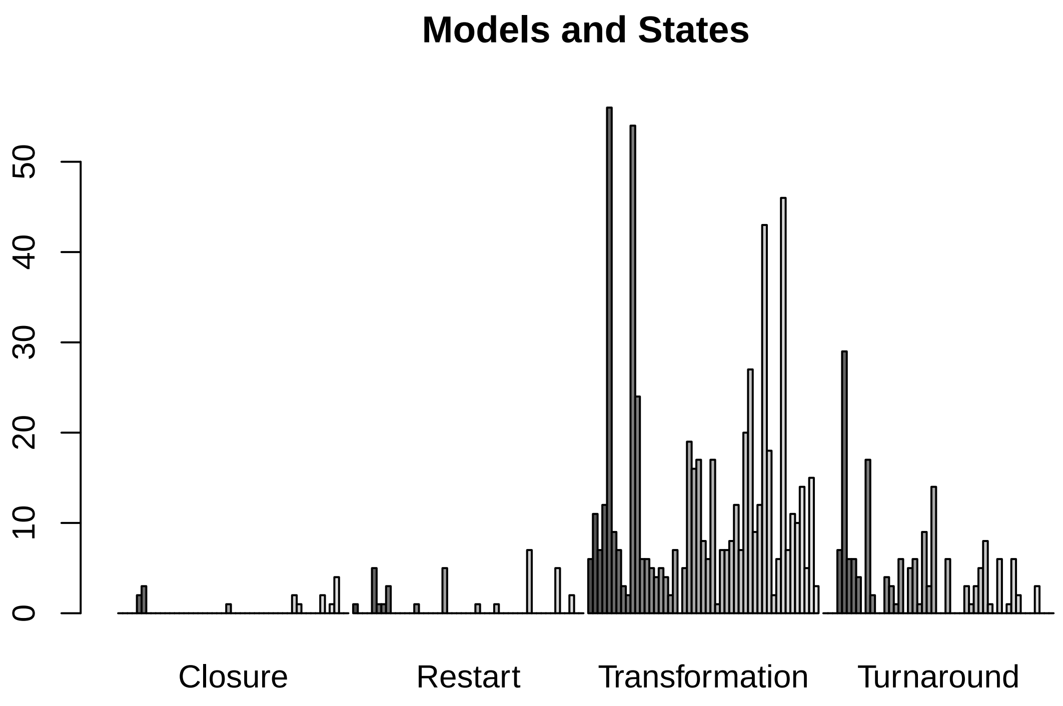
Part A

Data pre-processing

```
load('school-improvement-2010.rda')
# Filter out selected columns
df <- mydat[,c('State', 'Model.Selected')]
# Frequency table of State vs. Model.Selected
mytable <- table(df$State, df$Model.Selected)
# Filter out rows & columns of 0s only
mytable <- mytable[c(1:38, 40:50),2:5]
```

Some exploratory plots

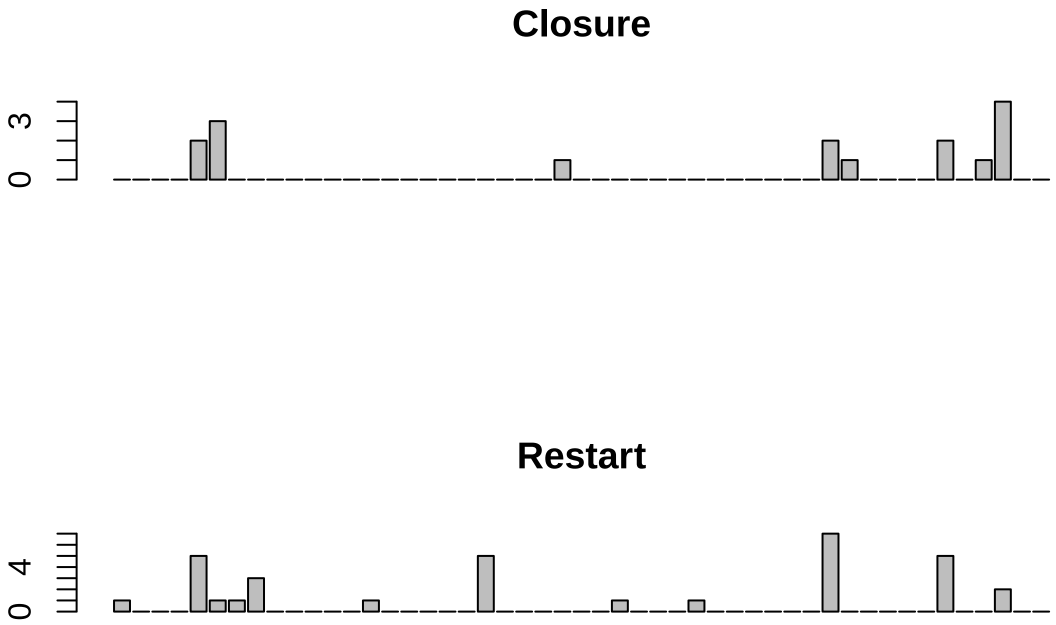
```
# Plot: Overview comparison across the 4 models
par(mfrow = c(1,1))
barplot(mytable, beside = T, main = 'Models and States')
```

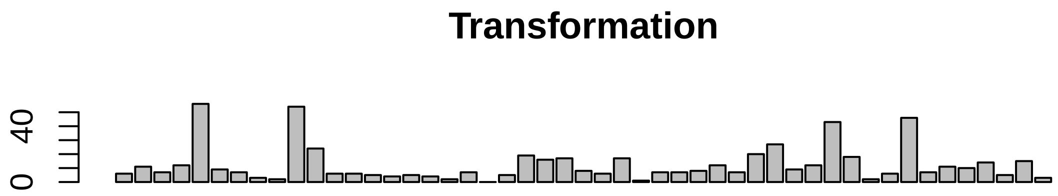


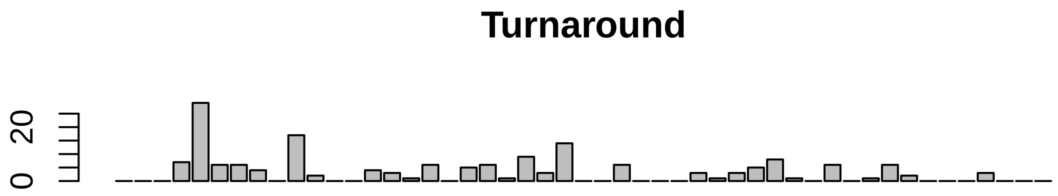
Plots: per Model.Selected

```
# Remove names
model <- colnames(mytable)
colnames(mytable) <- NULL
rownames(mytable) <- NULL
```

```r
par(mfrow = c(2,1))
for (i in 1:4) {
  temp <- mytable[,i]
  barplot(temp, main = model[i])
}
```

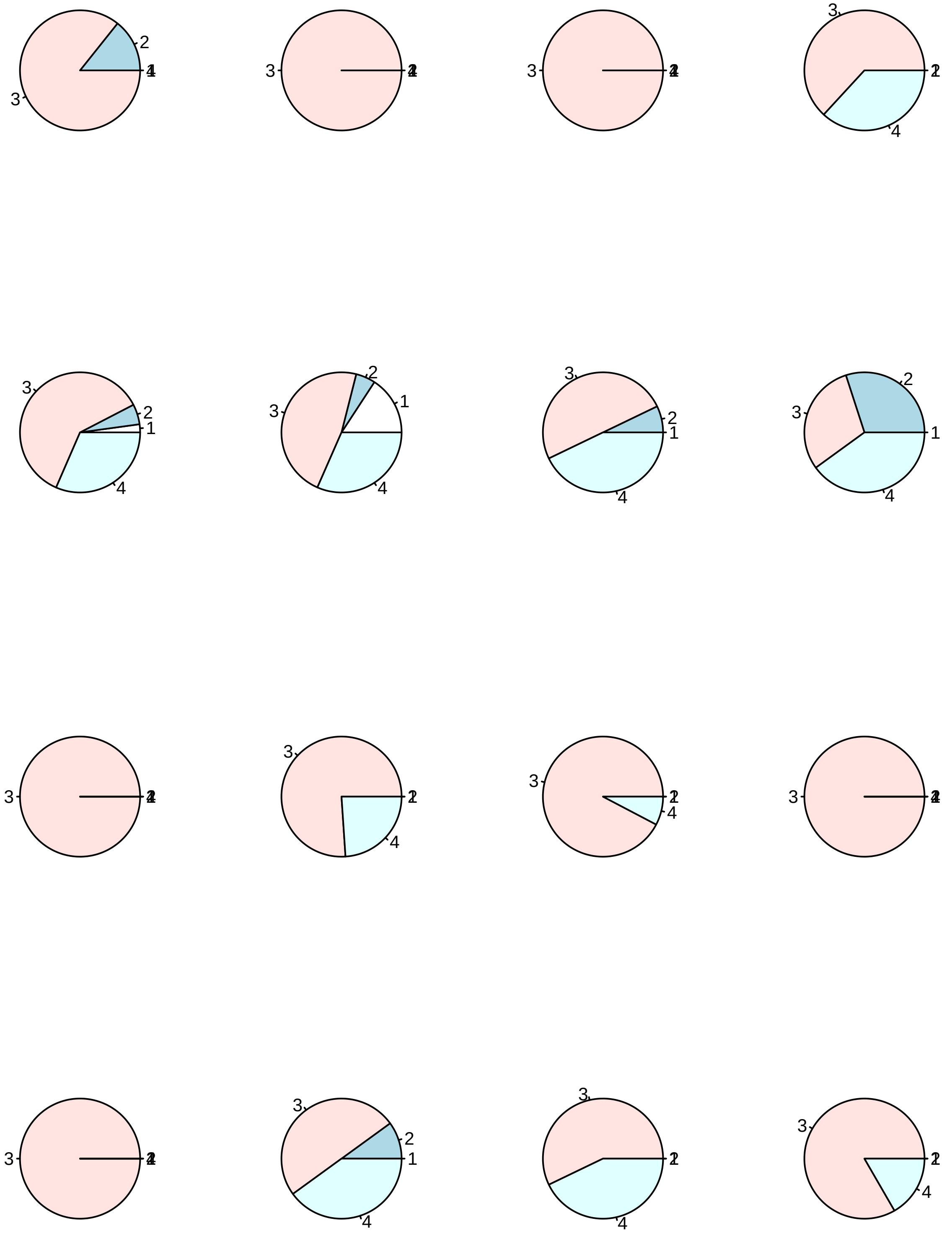






Plots: per State, for 16 random states

```r
par(mfrow = c(2,4))
k <- round(runif(16,1,49),0)
for (i in 1:16) {
  k <- k[i]
  temp <- mytable[i,]
  pie(temp, radius = 1)
}
```


2
1
4
3
3
2
4
3
2
4
3
1
2
4
3
2
1
4
3
2
1
4
3
2
1
4
3
2
1
4
3
2
4
3
1
2
4
3
1
2
4
3
1
2
4
3
2
4
3
2
1
4
3
1
2
4
3
1
2
4

Comments

From the barplot above, there seems to be differences in the distributions of models versus the states where the schools are located.

Hypothesis Test:

$$H_0 : \text{No Association: the model selected and the state are independent}$$

$$H_1 : \text{Association: the model selected and the state are not independent}$$

Chi-squared test:

```r
# Chi-squared test
chisq.test(mytable)
```

```

	Pearson's Chi-squared test

data:  mytable
X-squared = 378.37, df = 144, p-value < 2.2e-16
```

Comments

Given the p value of $2.2e-16$, which is extremely small, for any reasonable significance level such as $\alpha = .05$, we would reject H_0: there appears reasons that the model selected and the state where the school is located are not independent, and thus there is an association.

Part B

Yes, it is applicable to use calibration by permutation in this case.

```r
# Apply calibration by permutation
chisq.perm.test(mytable)
```

```
[1] 0.0004997501
```

Comments

The p value is now 0.0004997501, which is considerably larger than the p value from Chi-squared test. However, the p value is still considerably small, meaning for any reasonable significance level such as $\alpha = .05$, we would reject H_0: there appears reasons that the model selected and the state where the school is located are not independent, and thus there is an association.