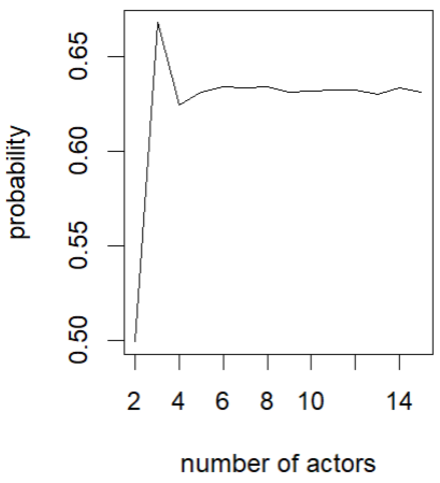
Module Code: CSMMS
Assignment report Title: ASSIGNMENT 2
Student Number: 32824514
Actual hours spent on the assignment: 20
Which Artificial Intelligence tools used (if applicable): None

Question 7: Famous Film Actors Matching Ratio

The following R codes are for calculating the probabilities and drawing the figure.

```
# 7 famous film actors matching ratio
# calculate the probabilities by 100,000 random experiments
cal_prop <- function(n){
   r <- 100000
   count <- 0
   for (i in 1:r) {
      sp <- sample(1:n)
      is.matched <- 0
      for (i in seq_along(sp)) {
         if (i == sp[i]) {
            is.matched <- 1
            break
         }
      }
      count <- count + is.matched
   }
   count / r
}

# Run the function from 2 to 15 to get the probabilities
ls <- list()
for (i in 2: 15) {
   ls[[i-1]] <- cal_prop(i)
}
plot(2: 15, unlist(ls), type='l',
        xlab='number of actors', ylab='probability')
```



As the figure shows, the number of probabilities starts at 0.5 at 2 actors and increases to reach its peak of about 0.67 at 3 actors. Then it fell back and hovered at 0.63. That means as the value gets larger, the number of full permutations of the new element and the number of all mismatched permutations grow at the same rate.

Question 8: Eigenfaces

Part A: Calculate the average of the three faces.

```
# 8 Eigenfaces
# 8.a calculate the average of the 3 faces
library(bmp)
img8.1 <- read.bmp(choose.files())
img8.2 <- read.bmp(choose.files())
img8.3 <- read.bmp(choose.files())
img8.avg <- (img8.1 + img8.2 + img8.3)/3
rotate <- function(x) t(apply(x, 2, rev))
image(rotate(img8.avg),col = gray((0:256)/256), axes=F)
```

The output figure of the codes:

Part B: Average Face

The codes following are to show the difference between each face to the average face.

```
# 8.b the difference of each face from the average face
img8.1.diff <- img8.1 - img8.avg
img8.2.diff <- img8.2 - img8.avg
img8.3.diff <- img8.3 - img8.avg
par(mfrow=c(1,3))
image(rotate(img8.1.diff + 128), col = gray((0:256)/256), axes=F, asp=1)
image(rotate(img8.2.diff + 128), col = gray((0:256)/256), axes=F, asp=1)
image(rotate(img8.3.diff + 128), col = gray((0:256)/256), axes=F, asp=1)
```

The output figure of the codes:



Part C: Eigenfaces

The codes following are to compute the eigenfaces.

```
# 8.c eigenfaces
```

```
# convert images to vectors and combine them
img8.diff.vec <- cbind(as.vector(img8.1.diff),
                       as.vector(img8.2.diff),
                       as.vector(img8.3.diff))

# calculate the covariance matrix
cov8.matrix <- cov(t(img8.diff.vec))

# calculate eigenvectors
eigen8.vectors <- eigen(cov8.matrix)$vectors

# convert vectors to matrices
row8 <- nrow(img8.1)
img8.1.eigenface <- matrix(eigen8.vectors[,1], nrow = row8, byrow = F)
img8.2.eigenface <- matrix(eigen8.vectors[,2], nrow = row8, byrow = F)
img8.3.eigenface <- matrix(eigen8.vectors[,3], nrow = row8, byrow = F)

# plot eigenfaces
par(mfrow=c(1,3))
image(rotate(img8.1.eigenface), col = gray((0:256)/256), axes=F, asp=1)
image(rotate(img8.2.eigenface), col = gray((0:256)/256), axes=F, asp=1)
image(rotate(img8.3.eigenface), col = gray((0:256)/256), axes=F, asp=1)
```

The output figure:

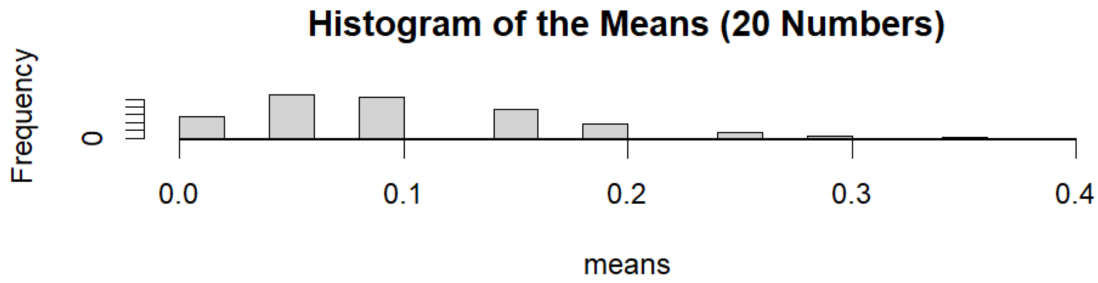
Question 9: Binomial Distribution datasets

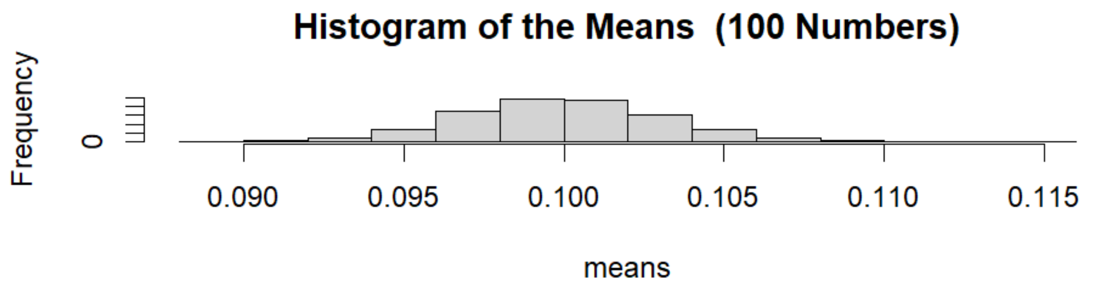
The following codes are for generating datasets and drawing distributions of means.

```
# 9 Binomial Distribution
# define generating function
generate_binomial_samples <- function(m, n) {
  colMeans(matrix(rbinom(n=m * n, size=10, prob=0.01), ncol = m, byrow=T))
}
# generate means
means9.1 <- generate_binomial_samples(10000, 20)
means9.2 <- generate_binomial_samples(10000, 100)
# plot figures
par(mfrow=c(2,1))
```

```
hist(means9.1, main='Histogram of the Means (20 Numbers)')
hist(means9.2, main='Histogram of the Means    (100 Numbers)')
```

The output figures:

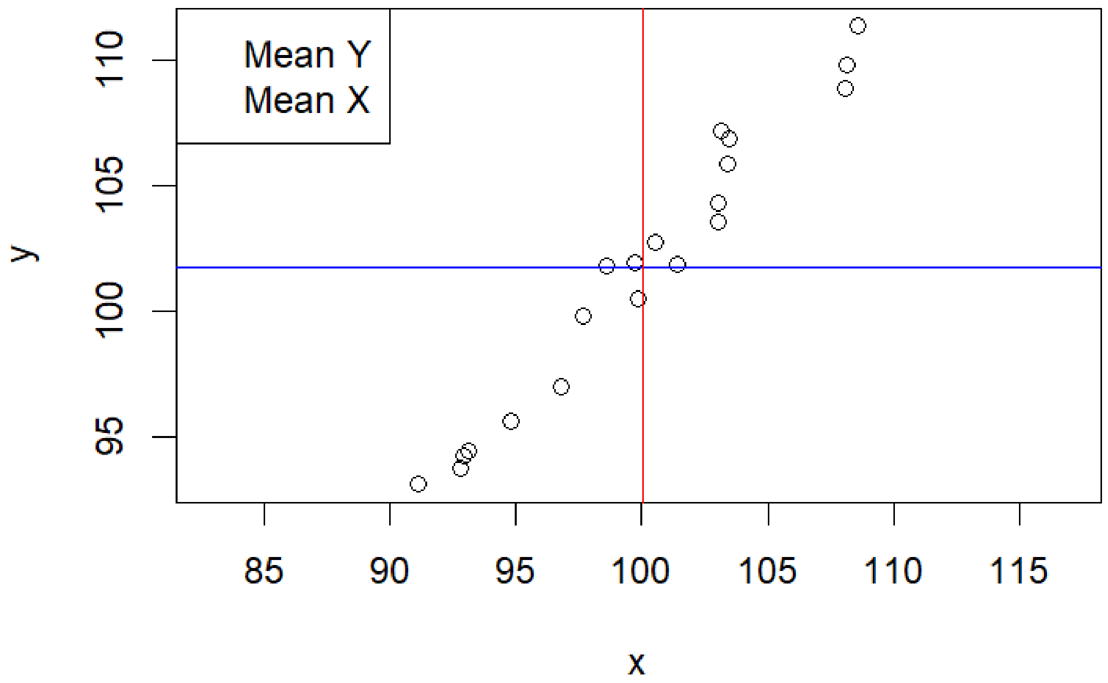




As the figures shown above, as the number increases from 20 to 100, the means change from scattered distribution to the bell-shaped (that is, normal) curves distribution with the center line at 0.1. This follows from the central limit theorem that the sum of many independent random variables has an approximately normal distribution. For the binomial probability model, the expectation: $E(Y) = n * prob = 10 * 0.01 = 0.1$ and the variable is: $Var(Y) = n * prob * (1 - prob) = 10 * 0.01 * (1 - 0.01) = 0.099$.

Question 10: Normal Distribution

```
# generate normal distribution numbers
set.seed(666)
x10 <- rnorm(20, mean=100, sd=4)
# generate noises
noise10 <- rnorm(length(x10), mean = 0, sd=1)
# calculate y values
y10 <- 2 + x10 + noise10
# plot scattered points
plot(y10~x10, asp=1, xlab='x', ylab='y')
# add mean lines
abline(h=mean(y10), col='blue')
abline(v=mean(x10), col='red')
legend('topleft', legend = c('Mean Y', 'Mean X'), col = c("blue", "red"))
```



Two-sample t test:

```
# two sample t test of X and Y
t.test(x10,y10,conf.level=0.05)
```

The output is:

```
        Welch Two Sample t-test

data:   x10 and y10
t = -1.1517, df = 37.852, p-value = 0.2567
alternative hypothesis: true difference in means is not equal to 0
5 percent confidence interval:
  -1.861848 -1.668356
sample estimates:
mean of x mean of y
  100.5437    102.3088
```

As the results shown above, the mean of x is 100.5437 and the mean of y is 102.3088. There is significant difference of value 102.3088-100.5437=0.7651. And they are also different from the theoretical means of 100 and 102.

Matched pairs t test:

```
# matched pair t test
t.test(y10-x10, conf.level = 0.05)
```

The output is:

```
        One Sample t-test

data:   y10 - x10
t = 7.2787, df = 19, p-value = 6.622e-07
alternative hypothesis: true mean is not equal to 0
```

```
5 percent confidence interval:
  1.749693 1.780511
sample estimates:
mean of x
  1.765102
```

As the results shown above, the mean of y-x is 1.765102, not 0. And the 5% confidence interval is 1.749693, 1.780511.

For this case, the matched pairs t test is more suitable for this data. Matched pairs t test is used for random pairs of dependent samples to account for the dependency. In this data, y depends on x (y=2+x+noise). Thus, matched pairs t test is better. In the other hand, two-sample t test is used for two independent random samples.

Question 11: Weighted Ridge Regression

Part a: Multiple linear regression to Weighted Ridge Regression

For a special vector **y**:

$$\boldsymbol{y}_{WR} = \left(y_1\sqrt{\omega_1}, y_2\sqrt{\omega_2}, \dots, y_n\sqrt{\omega_n}, 0, \dots, 0\right)^T = \left(\boldsymbol{y_{LS}} \cdot \sqrt{\boldsymbol{\omega}}, \boldsymbol{0_{p+1}}\right)^T,$$

and a special augmented matrix **X**:

$$\boldsymbol{X}_{WR} = \begin{pmatrix} \sqrt{\omega_1} & x_{1,1}\sqrt{\omega_1} & \dots & x_{1,p}\sqrt{\omega_1} \\ \dots & \dots & \dots & \dots \\ \sqrt{\omega_n} & x_{n,1}\sqrt{\omega_n} & \dots & x_{n,p}\sqrt{\omega_n} \\ \sqrt{\lambda} & 0 & 0 & 0 \\ 0 & \sqrt{\lambda} & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \sqrt{\lambda} \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{X}_{LS} \cdot \sqrt{\boldsymbol{\omega}} \\ \sqrt{\lambda}\boldsymbol{I}_{p+1} \end{pmatrix},$$

the expression can be simplified to

$$(\boldsymbol{y}_{WR} - \boldsymbol{X}_{WR}\boldsymbol{\beta})^T(\boldsymbol{y}_{WR} - \boldsymbol{X}_{WR}\boldsymbol{\beta})$$

$$= \sum_{i=1}^{n}\left(y_i\sqrt{\omega_i} - \beta_0\sqrt{\omega_i} - \sum_{j=1}^{p} x_{i,j}\sqrt{\omega_i}\beta_j\right)^2 + \sum_{i=n+1}^{m}\left(y_i - x_{i,i-n}\beta_{i-n}\right)^2$$

$$= \sum_{i=1}^{n} \omega_i\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \sum_{i=n+1}^{n+p}\left(0 - \sqrt{\lambda}\beta_{i-n}\right)^2$$

$$= \sum_{i=1}^{n} \omega_i\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \sum_{i=1}^{p}\left(\sqrt{\lambda}\beta_i\right)^2$$

$$= \sum_{i=1}^{n} \omega_i\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}$$

That means the weighted ridge regression estimates can be obtained from the multiple linear regression estimates by some transformation on **X** and **y**.

Thus, for the weighted ridge regression estimates

$\hat{\beta}^{WR} = (\boldsymbol{X}_{WR}^T\boldsymbol{X}_{WR})\boldsymbol{X}_{WR}^T\boldsymbol{y}_{WR}$.

Part b: Produce the weighted ridge estimates

```
# 11 Weighted ridge regression estimates
# 11.b function for the weighted ridge regression estimates
weighted_ridge_regression <- function(x, y, w, lambda) {
   w.sq <- sqrt(w)
   j <- ncol(x) + 1
   x.wr <- rbind(cbind(w.sq,x*w.sq), lambda*diag(j))
   y.wr <- matrix(c(y*w.sq, rep(0, j)), ncol=1, byrow=F)
   # t(x.wr) %*% x.wr %*% t(x.wr) %*% y.wr
   coef(lm(y.wr ~ x.wr))
}
# test the weighted ridge regression estimates
weighted_ridge_regression(
   x=matrix(c(1,2,3,4,8),nrow=5,byrow = T),
   y=c(1,2.1,3.05,3.95,8.9),
   w=c(1),
   lambda=1)
x11.1 <- c(1:100)
x11.2 <- sample(1:10, length(x11.1), replace = TRUE)
noise11 <- rnorm(length(x11.1), mean = 0, sd=0.05)
y11 <- 2 * x11.1 + 3 * x11.2 + noise11
weighted_ridge_regression(
   x=matrix(data = cbind(x1=x11.1, x2=x11.2), ncol = 2, byrow = F),
   y=y11,
   w=c(1,1),
   lambda=1)
```

Question 12: Multiply Questions
Part A: Shy Students
Part A.a: Where is Bo more likely to come from

We assume there are m Math-PhD students and b Business students. The matrix of the distribution of students is as follows:

	Math-PhD	Business	Total
Shy	0.35 * m	0.1 * b	0.35 * m + 0.1 * b
No-Shy	0.65 * m	0.9 * b	0.65 * m + 0.9 * b
Total	m	b	m + b

The probability that Bo is a Math-PhD student is

$$P(M|S) = \frac{P(M \cdot S)}{P(S)} = \frac{0.35*m}{0.35*m+0.1*b}.$$

The probability that Bo is a business student is

$$P(B|S) = \frac{P(B \cdot S)}{P(S)} = \frac{0.1*b}{0.35*m+0.1*b}.$$

As the numbers of students are unknown, we cannot exactly judge which one is more probable. If we assume that the numbers of students are arbitrary, then the expected numbers of students are the same, then it is more probable Bo is a Math-PhD student.

Part A.b: Where is Bo more likely to come from(2)

If the ratio of total number of students is 2:11, then the probability that Bo is a Math-PhD student is

$P(M|S) = \frac{0.35*m}{0.35*m+0.1*b} = \frac{0.35*2}{0.35*2+0.1*11} = \frac{7}{18}$.

The probability that Bo is a business student is

$P(B|S) = \frac{0.1*11}{0.35*2+0.1*11} = \frac{11}{18} > \frac{7}{18} = P(M|S)$.

Thus, Bo is more probable a business student.

Part B: Climb a staircase

We let w(n) be the number of all possible distinct ways of climbing a staircase with n steps in total to reach the top. Obviously, w(1)=1, w(2)=2, w(3)=4. For cases of $n \geq 4$, we break down it to the last step is a triple, a double and a single, then we have $w(n) = w(n-3) + w(n-2) + w(n-1)$. The codes following help us to compute the number of w(n).

```
# 12.B Climbing a staircase
ways12 <- function(n) {
  # for cases that n <= 3, return the results
  if (n <= 0) {
    return(0)
  }else if (n == 1) {
    return (1)
  }else if (n == 2) {
    return (2)
  }else if (n == 3) {
    return (4)
  }
  # for cases that n > 3, use a dynamic programming method
  # init a list to store the number of possible ways
  ways <- c(c(1,2,3),rep(0,n-3))
  # compute the values step by step
  for (i in 4:n){
    ways[[i]] <- ways[[i-3]] + ways[[i-2]] + ways[[i-1]]
  }
  print(ways)
  ways[[length(ways)]]
}
ways12(30)
```

```
 [1]       1       2       3       6      11      20      37      68     125     230
[11]     423     778    1431    2632    4841    8904   16377   30122   55403  101902
[21]  187427  344732  634061 1166220 2145013 3945294 7256527 13346834 24548655 45152016
```

Part B.a: climbing a staircase with a triple ending

The number of possible ways to climb 27 steps is $w(27) = 7256527$.

The number of possible ways to climb 30 steps is $w(30) = 45152016$.
Thus, the probability that Harry finished climbing by taking a triple is:

$$p = \frac{w(27)}{w(30)} = \frac{7256527}{45152016} \approx 0.16071.$$

Part B.b: multiple conditions climbing a staircase

There are w(17) possible ways to climb the beginning 17 step.
There is 1 possible way to climb a double from 18^{th} to 20^{th} step.
There is w(8) possible way to climb from 20th to 28th step.
There is 1 possible way to climb a triple from 28^{th} to 30^{th} step.
Thus, the number of possible ways following the rules above is:
$w_{condition} = w(17) * 1 * w(8) * 1 = 16377 * 1 * 68 * 1 = 1113636.$
The probability that Harry following all the rules above is:

$$p = \frac{w_{condition}}{w(30)} = \frac{1113636}{45152016} \approx 0.02466.$$

Part C: Defaulted Students

Part C.a: compare average income of defaulted and normal students

```
# 12.C.a: compare average income of defaulted and normal students
customers <- read.csv(file.choose())
head(customers)
avg_income_defaulted <-
mean(customers[customers$default=='Yes'&customers$student=='Yes',]$income)
avg_income_normal <-
mean(customers[customers$default!='Yes'&customers$student=='Yes',]$income)
sprintf("The average income of defaulted students is: %.2f",
        avg_income_defaulted)
sprintf("The average income of normal students is: %.2f",
        avg_income_normal)
if (avg_income_defaulted > avg_income_normal) {
  sprintf("The average income of defaulted students is %.2f%% higher than normal
students",
          (avg_income_defaulted-
avg_income_normal)/avg_income_normal*100)
} else if (avg_income_defaulted == avg_income_normal){
  print("The average income of defaulted students is equal to the one of normal
students")
} else {
  sprintf("The average income of normal students is %.2f%% higher than defaulted
students",
          (avg_income_normal-
avg_income_defaulted)/avg_income_defaulted*100)
}
```

The output of these codes above is:

```
[1] "The average income of defaulted students is: 17935.02"
[1] "The average income of normal students is: 18070.53"
[1] "The average income of normal students is 0.76% higher than defaulted students"
```

Part C.b: Logistic Regression

The following codes are for setting up the logistic regression.

```
# 12.C.b: logistic regression
# Fit logistic regression model
customers$default <- ifelse(customers$default=='Yes', 1, 0)
customers$student <- ifelse(customers$student=='Yes', 1, 0)
# train the model
lr.model <- glm(default ~ student + balance + income,
                data = customers,
                family = binomial)
lr.model$coefficients
predict(lr.model,data.frame(student=1, income=17500, balance=970),
type="response")
predict(lr.model,data.frame(student=1, income=17500, balance=970*1.04),
type="response")
```

The output:

```
   (Intercept)        student        balance         income
-1.030891e+01 -8.298770e-01  5.617580e-03 -9.841866e-06
0.002837557
0.003526179
```

That means the default probability for the students with (income, balance) = (17500, 970) is 0.00284. As their balance increases by 4% next year, the default probability will increase $\frac{0.00353-0.00284}{0.00284} \approx 24\%$ to be 0.00353.

Manual calculation:

The fitted model on the logit scale is:

$$logit(\hat{p}(x)) = \begin{cases} -10.309 - 0.830 + 0.0056 * bal - \frac{9.8}{10^6} * inc, & if\ student \\ -10.309 + 0.0056 * bal - \frac{9.8}{10^6} * inc, & if\ not\ student \end{cases}$$

The default probability for a student is:

$\hat{p}(x) = 1 - \frac{1}{1+e^{-11.139+0.0056*bal-0.0000098*inc}}$.

The default probability for a student with (income, balance) = (17500, 970) is

$\hat{p}(x) = 1 - \frac{1}{1+e^{-11.139+0.0056*970-0.0000098*17500}} = 0.00279$.

The partial difference in income=17500 is:

$\frac{d\hat{p}}{dbal} = \frac{1}{(1+e^{-11.3105+0.0056*bal})^2} * e^{-11.3105+0.0056*bal} * 0.0056$.

As the balance increases 4% from 970, the default probability will increase:

$$\Delta\hat{p} = \frac{1}{(1+e^{-11.3105+0.0056*970})^2} * e^{-11.3105+0.0056*970} * 0.0056 * (970 * 0.04)$$

$$= 0.000605.$$

That means the default probability for the students with (income, balance) = (17500, 970) is 0.00279. As their balance increases by 4% next year, the default probability will increase $\frac{0.000605}{0.00279} \approx 22\%$ to be 0.00340(The deviation between the result and the code calculation result is caused by the calculation accuracy).