FIT5197 Assignment 2 Semester 1, 2020

Do & Don't

1. Some of these policies are similar to your assignment 1; however, please read carefully as there are some differences between this assignment and the first assignment.
2. These questions are meant for you to solve independently, we encourage students to figure out the questions themselves as it would be good for their understandings of the topics; however, please feel free to consult your tutors if needed. Plagiarism (either from using online sources or copying the answers from your classmates) will be penalised accordingly.
3. Requests for special consideration or extension must be submitted at least 2 days BEFORE THE DEADLINE. The due date is on Sunday, so the latest day you can ask for extension is on Friday (the last official working day of the week for the teaching team). Please follow Monash guidelines to request for extensions (medical certificates, doctor or GP letter, etc). Emergencies are to be adjusted individually.
4. Please show all working when answering questions, you will not get full marks for a question if you don't comply.
5. Assignments need to be submitted in PDF and ipynb file format. Failure to comply will result in 20% penalty on each missing file.
6. Filename format for submitting the assignment should be “Assignment2_StudentId.pdf” and "Assignment2_StudentId.ipynb". Files with the wrong format incurs 20% penalty each.
7. This assignment has 10 marks for presentation, this includes presenting your explanation in **Markdown**, writing and commenting on code efficiently, creating good plots with clear labels on the axis, etc.
8. Only answers with correct methodology will be considered for consequential marks. Meaning if you attempt the question and your answer is incorrect, but your methodology is correct, you will still receive partial marks for subsequent questions. However, answers with incorrect methodology (misunderstanding the questions) will generate no marks for subsequent questions.
9. Challenge questions are for students aiming to get a HD for the assignment. We don't advise for students to spend time on these questions before finishing all the other parts in the assignment. This assignment is designed in a way that students can get up to 80 (HD) without attempting the challenge questions.
10. Please don’t send emails to tutors asking for suggestions, we have Moodle and consultations for that. When writing your inquiries on Moodle please try to be clear in your problem and not revealing your working to others as this might be counted as plagiarism on your part. A good format for inquiry topic would be e.g. “Assignment 2 – Tutorial 10 (your tutorial slot) – Question about median"
11. Handwritten answers incur a penalty of 10% on your assignment, you have Markdown, please learn how to use it as it will be an useful skill for you going through the degree as well as in real life situation.
12. This assignment will contribute towards 20% of your total score.
13. Late submission is 5% per day, after 10 days you will be given no marks. Late submission is calculated as follows: If you get 70% on this assignment and you are late for 2 days (you submit on Tuesday), your score is now 70% -10% (2x5% per day) = 60%. This is done to ensure that the teaching team can release your result as soon as possible so that you can review on your mistakes and have a better study experience.
14. Assignments shall be marked completely in two weeks’ time according to Monash Policies. If there are any changes to the marking time, we will duly inform you. Solutions will not be released for this assignment; you can come to the tutorial and ask for explanation about how to solve the questions after scores are released.

Question 1 - Probabilities (10 Marks)

A box contains n pairs of shoes ($2n$ shoes in total). If $2r$ (with the assumption that $2r \leq n$) shoes are selected at random, find the probability for the following scenarios:

Question 1a. (2 Marks)

A_0 = ‘No matching pair’

Answers 1.a

2r shoes are selected at random.

possible outcomes = choosing 2r shoes form 2n shoes

possible outcomes => $\binom{2n}{2r}$

To not have matching pairs, we can choose in $\binom{n}{2r}$ ways.

For each pair of shoes, there is a left and right shoe which can be chosen in 2^{2r} ways.

Thus our sample selection can be done in $\binom{n}{2r} * 2^{2r}$ ways.

p(‘No matching pair’) = $\frac{Sampleselection}{Totalpossibleoutcomes}$

$$P(A_0) = \frac{\binom{n}{2r} * 2^{2r}}{\binom{2n}{2r}}$$

Question 1b. (3 Marks)

A_1 = ‘only one matching pair’

Answers 1.b

We have n pairs of shoes.

first select one matching pair from n in $\binom{n}{1}$ ways.

Then, we have left n-1 pairs from which we select 2r-2 distinct pairs of shoes.

=> $\binom{n-1}{2r-2}$

shoes can be either left or right after the first matching pair. we can do this in 2^{2r-2} ways.

possible outcomes => $\binom{2n}{2r}$

$$P(A_1) = \frac{\binom{n}{1} * \binom{n-1}{2r-2} * 2^{2r-2}}{\binom{2n}{2r}}$$

Question 1c. (2 Marks)

A_2 = 'exactly two matching pairs'

Answers 1.c

Exactly 2 are matching in $\binom{n}{2}$ ways. Rest are n-2 pairs from which we select 2r-4 shoes.

so this can be dome in $\binom{n-2}{2r-4}$ ways.

shoes can be left or right, which can be selected in 2^{2r-4} ways.

$$P(A_2) = \frac{\binom{n}{2} * \binom{n-2}{2r-4} * 2^{2r-4}}{\binom{2n}{2r}}$$

Question 1d. (3 Marks)

A_r = 'exactly r matching pairs'

Answers 1.d

if exactly r matching pairs then we can do it in $\binom{n}{r}$ ways.

Rest are n-r pairs, and we are choosing 2r-2r = 0 shoes, so $\binom{n-r}{0}$ ways = 1 way.

shoes are left or right, can be done in 2^{2r-2r} ways = 0 ways.

so, $P(A_r) = \frac{\binom{n}{r}}{\binom{2n}{2r}}$

Question 2 - Conditional Probabilities & Entropy (30 Marks)

Warning, no built in functions to calculate probability or entropy from R should be used for this part. The only help you can get from R should be dataframe manipulation. Answers using functions will not be marked even if the answer is correct.

Sports analytics (i.e., the application of data science techniques to competitive sports) is a rapidly growing area of data science. In this question we will look at some very basic analytics applied to the outcomes of consecutive games of English Premier League (EPL). The file chelsea.csv contains a record of the outcomes of games of EPL played by **Chelsea football club (CFC)** in the seasons from 1993 to 2018. The data is sequential, in the sense that each row recorded the result whether the home team wins (H), the away team wins (A), or there is a draw (D).

Please show all working including code and presentation for this question

Part 1: Analyzing Home/Away performance (17 Marks)

Question 2.a (3 Marks)

Find out the probabilities **P(Chelsea Wins), P(Chelsea Loses), and P(Chelsea Draws)**. This includes all the results both home and away.

Answers 2.a

In [35]:
```
# Answer to Q2.a

chelsea = read.csv('chelsea.csv')

#chelsea home wins
chelsea.home.win = chelsea[chelsea$home=='Chelsea'&chelsea$result=='H',]
nrow(chelsea.home.win)

#chelsea away wins
chelsea.away.win = chelsea[chelsea$away=='Chelsea'&chelsea$result=='A',]
nrow(chelsea.away.win)

#chelsea total wins
chelseatotalwins = nrow(chelsea.home.win)+nrow(chelsea.away.win)
chelseatotalwins

# p chelsea wins
p_chelseawins = chelseatotalwins/nrow(chelsea)
p_chelseawins

#chelseahomeloss
chelsea.home.losses = chelsea[chelsea$home=='Chelsea'&chelsea$result=='A',]
nrow(chelsea.home.losses)

#chelsea away losses
chelsea.away.losses = chelsea[chelsea$away=='Chelsea'&chelsea$result=='H',]
nrow(chelsea.away.losses)

#chelsea total losses
chelseatotallosses = nrow(chelsea.home.losses)+nrow(chelsea.away.losses)
chelseatotallosses

p_chelsea_loses = chelseatotallosses/nrow(chelsea)
p_chelsea_loses

#chelsea total games
nrow(chelsea)
chelsea_total_draws = 958 - 523 - 201
chelsea_total_draws

p_chelsea_draws = chelsea_total_draws/nrow(chelsea)
p_chelsea_draws
```

304

219

523

0.545929018789144

61

140

201

0.209812108559499

958

234

0.244258872651357

Answers 2.a

$P(ChelseaWins) = \frac{chelseatotalwins}{nrow(chelsea)}$

$P(ChelseaWins) = \frac{523}{958}$

$P(ChelseaWins) = 0.545929018789144$

$P(ChelseaLoses) = \frac{chelseatotallosses}{nrow(chelsea)}$

$P(ChelseaLoses) = \frac{201}{958}$

$P(ChelseaLoses) = 0.209812108559499$

$P(ChelseaDraws) = \frac{chelsea_total_draws}{nrow(chelsea)}$

$P(ChelseaDraws) = \frac{234}{958}$

$P(ChelseaDraws) = 0.244258872651357$

Question 2.b (6 Marks)

Find out the conditional probabilities:

1. **P(Chelsea Wins| Playing at Home)**
2. **P(Chelsea Wins| Playing away)**
3. **P(Chelsea Draws| Playing at Home)**
4. **P(Chelsea Draws| Playing away)**
5. **P(Chelsea Loses| Playing at Home)**
6. **P(Chelsea Loses| Playing away)**

Please make comparison and a general conclusion.

Answers 2.b

In [36]:
```
#Q2.b

#chelsea home wins
chelsea.home.win = chelsea[chelsea$home=='Chelsea'&chelsea$result=='H',]
nrow(chelsea.home.win)

#chelsea away wins
chelsea.away.win = chelsea[chelsea$away=='Chelsea'&chelsea$result=='A',]
nrow(chelsea.away.win)


#chelsea home losses
chelsea.home.losses = chelsea[chelsea$home=='Chelsea'&chelsea$result=='A',]
nrow(chelsea.home.losses)

#chelsea away losses
chelsea.away.losses = chelsea[chelsea$away=='Chelsea'&chelsea$result=='H',]
nrow(chelsea.away.losses)


#how many games chelsea plays at home
chelsea.playing.at.home = chelsea[chelsea$home == 'Chelsea',]
chelsea.playing.at.home
nrow(chelsea.playing.at.home)

# probability of chelsea playing at home
p_chelsea_playing_home = nrow(chelsea.playing.at.home)/nrow(chelsea)
p_chelsea_playing_home


#how may games chelsea plays away
chelsea.playing.away = 958-nrow(chelsea.playing.at.home)
chelsea.playing.away
nrow(chelsea.playing.away)

# probability of chelsea playing away
p_chelsea_playing_away = 1 - p_chelsea_playing_home
p_chelsea_playing_away


# games chelsea draws playing home
chelsea.draws.playing.home = chelsea[chelsea$home=='Chelsea'&chelsea$result=='D',]
nrow(chelsea.draws.playing.home)

#games chelsea draws playing away
chelsea.draws.playing.away = chelsea[chelsea$away=='Chelsea'&chelsea$result=='D',]
nrow(chelsea.draws.playing.away)
```

A data.frame: 479 × 3

Answers 2.b

P(Chelsea Playing at Home) = 0.5

P(Chelsea Playing away) = 0.5

1. **P(Chelsea Wins| Playing at Home)**

=> $P(\frac{Chelseawins \cap Playingathome}{playingathome})$

=> $\frac{304/958}{479/958}$

=> $\frac{304}{479}$

2. **P(Chelsea Wins| Playing away)**

=> $P(\frac{Chelseawins \cap Playingaway}{playingaway})$

=> $\frac{219/958}{479/958}$

=> $\frac{219}{479}$

2. **P(Chelsea Draws| Playing at Home)**

=> $P(\frac{ChelseaDraws \cap PlayingHome}{playingatHome})$

=> $\frac{114/958}{479/958}$

=> $\frac{114}{479}$

4. **P(Chelsea Draws| Playing away)**

=> $P(\frac{ChelseaDraws \cap Playingaway}{playingaway})$

=> $\frac{120/958}{479/958}$

=> $\frac{120}{479}$

5. **P(Chelsea Loses| Playing at Home)**

=> $P(\frac{ChelseaLoses \cap Playingathome}{playingathome})$

=> $\frac{61/958}{479/958}$

=> $\frac{61}{479}$

6. **P(Chelsea Loses| Playing away)**

=> $P(\frac{ChelseaLoses \cap Playingaway}{playingaway})$

=> $\frac{140/958}{479/958}$

=> $\frac{140}{479}$

Question 2.c (3 Marks)

Find $H(\text{Chelsea Results})$ this includes results in both home and away games

Answers 2.c

In [37]:
```
#  chelsea home- win,loss,draw and Away- Win.loss,draw probabilities for entropy calculation

entropy_df = rbind(c('L', 'W', 'D'), c(201/958,523/958,234/958))
as.data.frame(entropy_df)
```

A data.frame: 2 × 3

V1	V2	V3
<fct>	<fct>	<fct>
L	W	D
0.209812108559499	0.545929018789144	0.244258872651357

Manual calculations below:

pi -> index probability for each outcome L,W,D from previous R table.

$H(\text{Chelsea Results})$ = entropy of chelsea results.

$H(\text{Chelsea Results}) = \sum_{x=1}^{3}(pi * \log_2(1/pi))$

$H(\text{Chelsea Results})$ = 0.4727+0.4767+0.496 = 1.446

$H(\text{Chelsea Results})$ = 1.446 bits of information

Question 2.d (5 Marks)

Is knowing whether Chelsea plays at home or away a good indicator in knowing the result of CFC games? Show your justification (answering just yes or no will not be given any marks) by calculating all the information necessary using the knowledge you have learnt so far in the unit.

Note that this is a challenge question, only attempt if you are comfortable with your progress. You should not use consultations to ask for this question as the tutors will proritize answering queries about other questions.

In [38]:
```
# this is the joint entropy table for chelsea FC comprising of Home and away, lossses wins and draws.

new_entropy_df = rbind(c('HL', 'HW', 'HD', 'AL','AW','AD'), c(61/958,304/958,114/958,0,0,0),c(0,0,0,140/958,219
as.data.frame(new_entropy_df)
```

A data.frame: 3 × 6

V1	V2	V3	V4	V5	V6
<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
HL	HW	HD	AL	AW	AD
0.0636743215031315	0.317327766179541	0.118997912317328	0	0	0
0	0	0	0.146137787056367	0.228601252609603	0.125260960334029

Answers 2.d

Let, Y be a random variable that determines whether chelsea plays at home or away.

Let, R be the random variable that determines chelsea results(from previous calculations)

The above table shows the joint probability distributioin of random variables R (from previous entropy calculation) and Y.

previously we get:

$H(\text{Chelsea Results})$ = 1.446 bits of information = H(R)

from the joint probability distribution we can calculate the joint entropy of random variables R and Y.

H(R) = 1.446 bits

H(R,Y) = 2.41 bits

H(Y) = 0.5*logbase2(1/0.5)* + *0.5*logbase2(1/0.5) = 1 , (this is the entropy of home or away)

H(R|Y) = H(R,Y) - H(Y) (joint entropy used in the definition of conditional entropy)

H(R|Y) = 2.411 - 1 = 1.411 bits

Here we can see that, the conditional entropy (H(R|Y)) of CFC results given that we have knowledge of whether chelsea played home or away, is less than the joint entropy (H(R,Y)). This means by knowing whether chelsea plays at home or away, reduces the uncertainty of determining the result of chelsea's game. In other words, knowing whether chelsea is playing at home or away is a good indicator in knowing the result of CFC games.

Part 2: Analyzing effects of previous results on future results (13 Marks)

This is a new part of the question, this part will focus on a different aspects compared to part I. Objectively speaking, this part is harder compared to the previous one; thus, it has lower mark allocation, students are advised to spend time on this part if they want to achieve desirable outcome.

Based on the data given to you, please create another column named "binary". This column will record a win $(\text{corresponding to } 1)$ or a loss/draw $(\text{corresponding to } 0)$ in the order in which the games were played by CFC.

A simple question regarding this type of data might be regarding the existence of (de)motivativing effects on a team if they have won / not won their previous game. Let W_t denote the binary outcome of a game in round t and W_{t-1} denote the outcome of the game played in the previous round. Answer the following questions; **you must provide working/justification.**

Answers Part 2:

In [39]:
```
#part 2: analysing effects of previous result on future results

#create new column named binary with zeroes
chelsea["binary"] <- 0


#replacing binary column with either 1 or 0 based on condition
chelsea$binary <- ifelse(chelsea$home == "Chelsea"&chelsea$result=='H',1, chelsea$binary)

chelsea$binary <- ifelse(chelsea$home == "Chelsea"&chelsea$result=='A',0, chelsea$binary)

chelsea$binary <- ifelse(chelsea$home == "Chelsea"&chelsea$result=='D',0, chelsea$binary)

chelsea$binary <- ifelse(chelsea$away == "Chelsea"&chelsea$result=='A',1, chelsea$binary)

chelsea$binary <- ifelse(chelsea$away == "Chelsea"&chelsea$result=='H',0, chelsea$binary)

chelsea$binary <- ifelse(chelsea$away == "Chelsea"&chelsea$result=='D',0, chelsea$binary)



# to check if binary column is working, result is 523 rows which is the number of chelsea wins in home and away
# 435 is chelsea losses and draws
nrow(chelsea[chelsea$binary==1,]) # chelsea wins
nrow(chelsea[chelsea$binary==0,]) # chelsea losses and draws
```

523

435

Question 2.e (4 Marks)

Using the data in **chelsea.csv** and the new column you just created for this task, write R code to **find the frequency** with which CFC won / did not win a game after it won / did not win its previous game. Using these frequencies, calculate the joint distributions $P(W_t = 0, W_{t-1} = 0)$, $P(W_t = 1, W_{t-1} = 0)$, $P(W_t = 1, W_{t-1} = 1)$, and $P(W_t = 0, W_{t-1} = 1)$. We suggest students create another column from the original dataframe to solve this question. Please read this question carefully before attempting.

Answers 2.e

W_t = Binary outcome in rount t

W_{t-1} = Binary outcome in previous round (t-1)

Home chelsea and result is H = 1

Home chelsea and result is A = 0

Home chelsea and result is D = 0

In [40]:
```
#creating frequency column in chelsea data frame

chelsea["frequency"] <- 0
chelsea$frequency <- as.character(chelsea$frequency)

#updating frequency based on win/loss binary column

for(i in 2:nrow(chelsea)){
  if(chelsea$binary[i]==0&chelsea$binary[i-1]==0){
    chelsea$frequency[i]='00'
  }

  else if(chelsea$binary[i]==0&chelsea$binary[i-1]==1){
    chelsea$frequency[i]='10'
  }

  else if(chelsea$binary[i]==1&chelsea$binary[i-1]==0){
    chelsea$frequency[i]='01'
  }

  else if(chelsea$binary[i]==1&chelsea$binary[i-1]==1){
    chelsea$frequency[i]='11'
  }
}



# counts for conditional probabilities for joint probability distribution

# counts of consecutive not wins
count00 = subset(chelsea, chelsea$frequency=="00")
nrow(count00)

# counts of previous game not win, current game win
count01 = subset(chelsea, chelsea$frequency=="01")
nrow(count01)

#counts of previous game win, current game not win
count10 = subset(chelsea, chelsea$frequency=="10")
nrow(count10)

# counts of consecutive wins
count11 = subset(chelsea, chelsea$frequency=="11")
nrow(count11)


# joint distribution probabilities

p_00 = nrow(count00)/(nrow(chelsea)-1)
p_00
p_01 = nrow(count01)/(nrow(chelsea)-1)
p_01
p_10 = nrow(count10)/(nrow(chelsea)-1)
p_10
p_11 = nrow(count11)/(nrow(chelsea)-1)
p_11
```

198

236

236

287

0.206896551724138

0.246603970741902

0.246603970741902

0.299895506792059

Joint distributions

$P(W_t = 0, W_{t-1} = 0)$ **= 0.206896551724138**

$P(W_t = 1, W_{t-1} = 0)$ **= 0.246603970741902**

$P(W_t = 1, W_{t-1} = 1)$ **= 0.299895506792059**

$P(W_t = 0, W_{t-1} = 1)$ **= 0.246603970741902**

Question 2.f (2 Marks)

What is the probability that CFC will win a game given that they won their previous game?

Answers 2.f

Calculation of the probability when CFC wins a game given that they won their previous game too.

$P(W_t = 1 \mid W_{t-1} = 1) = \frac{P(W_t=1, W_{t-1}=1)}{P(W_{t-1}=1)}$

$P(W_{t-1} = 1)$ = 0.2466+0.2998 = 0.5464

$P(W_t = 1 \mid W_{t-1} = 1)$ = 0.2998/0.5464 = 0.5486

Question 2.g (2 Marks)

What is the probability that CFC will win a game given that they didn't win their previous game?

Answers 2.g

Calculation of the probability when CFC wins a game given that they did not win their previous game.

$P(W_t = 1 \mid W_{t-1} = 0) = \frac{P(W_t=1, W_{t-1}=0)}{P(W_{t-1}=0)}$

$P(W_{t-1} = 0)$ = 0.2068+0.2466 = 0.4534

$P(W_t = 1 \mid W_{t-1} = 0)$ = 0.2466/0.4534 = 0.5439

Question 2.h (2 Marks)

Do you think winning/not winning the previous game had an effect on the CFC players in their next game? Justify your answer? **(Note that this is different compared to 2.d)**

Answers 2.h

Interestingly enough, winning/not winning the previous game doesnt have much effect on CFC players in their next game. This is because the conditional probabilities of chelsea winning a game given that they won their previous game(0.5486) is very close to the conditional probability of chelsea winning a game given that they didnt win their previous game(0.5439). Thus a comment can be made about the mental aptitude of CFC players - they do not get demotivated for the next game even if they loose a game.

Question 2.i (3 Marks)

Calculate the probability of CFC not winning their next two games given that they won their previous game.

In [41]:
```
# newfrequency column for part 2.i
chelsea["newfrequency"] <- 0
chelsea$newfrequency <- as.character(chelsea$newfrequency)

#updating newfrequency based on win/loss binary column,
#for probability cfc winning 1 game, then not winning next 2

#'100' denotes that chelsea wins 1 game then doesnt win 2 game after that 1 win.
#'101' denotes that chelsea wins 1, doesnt win 1, then wins 1.
#'111' denotes that chelsea wins 3 consecutive games.
#'110' denotes that chelsea wins 2 games then doesnt win 1.

# im only creating values in newfreuency columns where chelsea wins atleast 1 game out of 3(only this much requ
#for answering question 2.i)

for(i in 3:nrow(chelsea)){
   if(chelsea$binary[i]==0&chelsea$binary[i-1]==0&chelsea$binary[i-2]==1){
    chelsea$newfrequency[i]='100'
   }
   else if(chelsea$binary[i]==1&chelsea$binary[i-1]==0&chelsea$binary[i-2]==1){
    chelsea$newfrequency[i]='101'
  }
   else if(chelsea$binary[i]==1&chelsea$binary[i-1]==1&chelsea$binary[i-2]==1){
    chelsea$newfrequency[i]='111'
  }
   else if(chelsea$binary[i]==0&chelsea$binary[i-1]==1&chelsea$binary[i-2]==1){
    chelsea$newfrequency[i]='110'
  }

}


#counts of chelsea not winning next 2 games after they won a game
count100 = subset(chelsea, chelsea$newfrequency=="100")
nrow(count100)

count101 = subset(chelsea, chelsea$newfrequency=="101")
nrow(count101)

count111 = subset(chelsea, chelsea$newfrequency=="111")
nrow(count111)

count110 = subset(chelsea, chelsea$newfrequency=="110")
nrow(count110)


#probabilities of 3 consequtive matches in cases of 110,101,111 and 110
p_100 = nrow(count100)/(nrow(chelsea)-2)
p_100
p_101 = nrow(count101)/(nrow(chelsea)-2)
p_101
p_111 = nrow(count111)/(nrow(chelsea)-2)
p_111
p_110 = nrow(count110)/(nrow(chelsea)-2)
p_110
```

104

132

163

124

0.108786610878661

0.138075313807531

0.170502092050209

0.129707112970711

Answers 2.i

Here i made a newfrequency column to count occurances where:

1.chelsea won a game then didnt win 2 games after that. '100'

2.chelsea wins 1, doesnt win 1, then wins 1. '101'

3.chelsea wins 3 consecutive games. '111'

4.chelsea wins 2 games then doesnt win 1. '110'

then calculated probability from those frequencies:

$P(CFCnotwinningnext2games \mid Theywontheirpreviousgame)$ =>

=> $\frac{P(CFCnotwinningnext2games \cap Theywontheirpreviousgame)}{P(Theywontheirpreviousgame)}$

Here won their previous game means -> won atleast the first game(out of 3 consequtive games)

=> $P(Theywontheirpreviousgame) = p(100) + p(101) + p(111) + p(110)$

=> $\frac{p(100)}{p(100)+p(101)+p(111)+p(110)}$

=> $\frac{0.1088}{0.1088+0.13807+0.1705+0.1297}$

=> $\frac{0.1088}{0.54707}$ = 0.1989

Question 3 - Expectation - Challenge Question (10 Marks)

Randomly place a point P inside triangle ABC. Let X be the continuous random variable representing the distance from point P to AB. Find out $E(X)$ & $Var(X)$ (**5 Marks each**)

Note that this is a challenge question, only attempt if you are comfortable with your progress. You should not use consultations to ask about this question as the tutors will prioritize answering queries about other questions.

Answers 3

h -> height of triangle ABC and also the distance from C to AB.

X -> is the distance from point P to AB which is a continuous random variable.

Probability density function (pdf) of X ->

$$pdf(x) = \begin{cases} 1/h & \text{if } 0 < x < h \\ 0 & \text{else.} \end{cases}$$

Expected value of X = E[X] = $\left(\int_0^h x(pdf(x))\ dx\right)$ =>

=> $\left(\int_0^h (x)(\frac{1}{h})\ dx\right)$

=> $(\frac{x^2}{2h})\Big|_0^h$

E(x) = $\frac{h^2}{2h}$ - 0

E(x) = $\frac{h}{2}$

On average distance X is half of height of triangle.

Expected value of X^2 = E[X^2] = $\left(\int_0^h x^2(pdf(x))\ dx\right)$ =>

=> $\left(\int_0^h (x^2)(\frac{1}{h})\ dx\right)$

=> $(\frac{x^3}{3h})\Big|_0^h$

E[X^2] = $\frac{h^3}{3h}$ - 0

E[X^2] = $\frac{h^2}{3}$

Var[X] = E[X^2] - [$E[X]$]^2

=> $\frac{h^2}{3}$ - $(\frac{h}{2})^2$

=> $\frac{4(h^2)-3(h^2)}{12}$

Var[X] = $\frac{h^2}{12}$

note: assumption of X as a uniform random variable.

Question 4 - Distribution (10 Marks)

The teaching team of FIT5197 is required to prepare 4 questions each week for the next week's tutorial. The number of questions created in a week is said to have a Poisson distribution with mean 6.

Question 4.a (2 Marks)

Find the probability that the teaching team manages to write enough questions for the following week?

In [42]:
```
#poisson dist q4.a

#P(X<4)
p_xless_than_4 = (6^0*exp(-6)/factorial(0))+(6^1*exp(-6)/factorial(1))+(6^2*exp(-6)/factorial(2))+(6^3*exp(-6)/
p_xless_than_4

#P(X>=4)
p_xlgreater_equal_4 = 1-p_xless_than_4
p_xlgreater_equal_4
```

0.151203882776648

0.848796117223352

Answers 4.a

Questions created in a week has a poisson distribution with mean of 6.

Let number of questions be a random variable X.

$X \sim$ Pois(6)

P(teaching team manages to write enough questions ofr the following week) = P($x \geq 4$)

P($x \geq 4$) = 1 - P($x < 4$) = 1 - 0.151203882776648 = 0.848796117223352

Question 4.b (4 Marks)

Since some of the tutors in the teaching team are also responsible for other units from FIT, for each week, there is a probability of 40% that only half of the team will work on the questions. If that is the case, the teaching team can only create 3 questions on average. If the teaching team fails to finish 4 questions one week, what is the probability that only half of the team works that week?

In [43]:
```
#4.b new dist y ~ pois(3)

p_yless_than_4 = (3^0*exp(-3)/factorial(0))+(3^1*exp(-3)/factorial(1))+(3^2*exp(-3)/factorial(2))+(3^3*exp(-3)/
p_yless_than_4

p_y = dpois(0,3)+dpois(1,3)+dpois(2,3)+dpois(3,3)
p_y


p_x_equal_to_5 = (5^5*exp(-5)/factorial(5))
p_x_equal_to_5
```

0.647231888782231

0.647231888782231

0.175467369767851

Answers 4.b

P(only half teaching team working on the questions) = 0.4 = $P(A)$

conditional questions created on the new information that half teaching team working can be distrubuted using a poisson distribution (Y) with mean 3.

$Y \sim$ Pois(3)

we need to find the probability: $P(halfteachingteamworking \mid failstofinish4questions)$ = $P(A \mid B)$

P(Y<4) = $P(Failingtofinish4Qs \mid halfteachingteam)$ = 0.6472319 = $P(B \mid A)$ P(X<4) = $P(Failingtofinish4Qs \mid Fullteachingteam)$ = 0.1512039

P(failing to finish 4 questions) = 0.4*0.6472*+*0.6*0.1512 = 0.3496 = $P(B)$

Using bayes theorem ->

$P(A \mid B) = \frac{P(B|A)*P(A)}{P(B)}$

$P(A \mid B) = \frac{0.6472319*0.4}{0.3496}$ = 0.7405

Question 4.c (4 Marks)

On week 12, the teaching team decides to no longer limit to 4 questions, and instead use every question they create. If a student has a 40% chance of correctly answering questions, and this student is expected to answer 2 questions correctly in the coming tutorial, what is the probability that the whole teaching team worked on creating the questions that week?

Answers 4.c

Now the teaching team is using all the questions they create.

P(correctly answering questions) = 0.4

Student correctly answer questions can be distributed as: $S \sim \text{Pois}(2)$

Let, total questions be x

number of correct answers = 0.4x

E[0.4x] = 2 , This means expected value of correct answers is 2

0.4E[x] = 2

E[x] = $\frac{2}{0.4}$ = 5 , Teaching team on avg will create 5 quesition that week

We need to find p(Whole team worked given x=5) = $P(A \mid B)$

$P(B \mid A) = P(x = 5 \mid wholeteamworked) = \frac{6^5 * e^{-6}}{5!}$ = 0.1606 (used X ~ pois(6))

$P(B \mid A)$ = 5 questions created given whole team worked

Using bayes theorem ->

$P(A \mid B) = \frac{P(B|A) * P(A)}{P(B)}$

P(B) = probability of creating 5 question by both half team and full team

need to also find, p(created 5 given half teaching team working)

Let, y ~ pois(3) , here using mean 3

p(y=5) = $\frac{3^5 * e^{-3}}{5!}$ = 0.1008188

P(B) = 0.4 * 0.1008188+ 0.6 * 0.1606 = 0.13668752

$P(A \mid B) = \frac{0.1606 * 0.6}{0.13668752}$ = 0.7049

So, required Probability from this question's answer is 0.7049

Question 5 - Maximum Likelihood Estimation (15 Marks)

The exponential distribution is a probability distribution for non-negative real numbers. It is often used to model waiting or survival times. The version that we will look at has a probability density function of the form

$$p(y|v) = exp(-e^{-v} y - v)$$

where $y \in R_+$, i.e., y can take on the values of non-negative real numbers. In this form it has one parameter: a log-scale parameter v. If a random variable follows a gamma distribution with log-scale v we say that $Y \sim Exp(v)$. If $Y \sim Exp(v)$, then $E[Y] = e^v$ and $V[Y] = e^{2v}$.

Question 5.a (4 Marks)

Imagine we are given a sample of n observations $y = (y_1, \ldots, y_n)$. Write down the joint probability of this sample of data, under the assumption that it came from an exponential distribution with log-scale parameter v (i.e., write down the likelihood of this data). Make sure to simplify your expression, and provide working.

Likelyhood of the given data

Likelyhood function = $\prod_{i=1}^{n} p(y_i|v)$

$P(Y \mid v) = \prod_{i=1}^{n} exp(-e^{-v} y_i - v)$

=> $\prod_{i=1}^{n} (exp(-v))(exp(-e^{-v} y_i))$

Likelyhood of the data = $(e^{-nv})(e^{-\sum_{i=1}^{n} y_i e^{-v}})$

Question 5.b (2 Marks)

Take the negative logarithm of your likelihood expression and write down the negative loglikelihood of the data y under the exponential model with log-scale v. Simplify this expression

Negative log likelyhood

Negative log likelyhood = $-\log((e^{-nv})(e^{-\sum_{i=1}^{n} y_i e^{-v}}))$

= $-(-nv - \sum_{i=1}^{n} y_i(e^{-v}))$

= $nv + \sum_{i=1}^{n} y_i(e^{-v})$

Question 5.c (4 Marks)

Derive the maximum likelihood estimator $\hat{v}$ for v. That is, find the value of v that minimises the negative log-likelihood. You must provide working.

Deriving the maximum likelyhood estimator $\hat{v}$

$\hat{v}$ = ?

We have to find the value of $\hat{v}$ where the negative log likelyhood is minimised.

so, we take the derivative of $nv + \sum_{i=1}^{n} y_i(e^{-v})$ and set it to 0 to find the maximum likelyhood estimator $\hat{v}$.

$\frac{d}{dv}(nv + \sum_{i=1}^{n} y_i(e^{-v}))$ = 0

=> $n + (\sum_{i=1}^{n} y_i(-e^{-v}))$ = 0

=> $n - (e^{-v}) \sum_{i=1}^{n} y_i$ = 0

=> $n = (e^{-v}) \sum_{i=1}^{n} y_i$

=> $e^{v} = \frac{\sum_{i=1}^{n} y_i}{n}$

=> $\frac{\sum_{i=1}^{n} y_i}{n} = \overline{y}$

=> $e^{v} = \overline{y}$

=> $\hat{v} = \log(\overline{y})$

maximum likelyhood estimator $\hat{v} = \log(\overline{y})$

Question 5.d (5 Marks)

Determine the approximate bias and variance of the maximum likelihood estimator $\hat{v}$ of v for the exponential distribution.

Note that this is a challenge question, only attempt if you are comfortable with your progress. You should not use consultations to ask about this question as the tutors will proritize answering queries about other questions.

Approximate bias and variance of the maximum likelihood estimator $\hat{v}$

Given n is sufficiently large, by following the asymptotic properties of maximum likelyhood estimators (namely the large sample property of MLE):

$\hat{v}$ is approximately $\hat{v} \sim$ N(v, $I^{-1}(v)$)

Here, the inverse varianc I(v) is the fishers information.

I(v) = Expected value of second derivative of $nv + \sum_{i=1}^{n} y_i(e^{-v})$ which is the negative log likelyhood function.

I(v) = e^{-v} E[($\sum_{i=1}^{n} y_i$)] = $e^{-v} * n * e^{v}$ = n

$I^{-1}(v) = \frac{1}{n}$, this is the inverse of I(v)

$\hat{v}$ is approximately $\hat{v} \sim$ N(v, $\frac{1}{n}$)

Approximate bias for $\hat{v}$ is E[$\hat{v}$ - v] $\approx$ 0

The approximate variance that we can find here is the inverse of the fishers information = $\frac{1}{n}$

Question 6 - Central Limit Theorem (15 Marks)

Sampling Process: Assume that we randomly select samples of the same size n an infinite number of times from a population that follows a Poisson distribution with mean of λ, and then, we calculate the mean of scores in each sample.

Question 6.a (2 Marks)

What does Central Limit Theorem tell us about the sampling distribution of the sample mean?

Answer 6.a

According to the Central Limit theorem (as long as random variables are independant), the sampling distribution of the sample means approaches a gaussian distribution as the sample size gets larger regardless of the shape of the population distribution. This is especially true when n is greater than 30. If we take more samples, the sample means graph will look like a gaussian distribution.

Question 6.b (3 Marks)

For three different Poisson populations with mean of λ_1 = 1, λ_2 = 5 and λ_3 = 20, we will do the sampling four separate times -- for small samples (n=10), for samples of 100 subjects (n=100) and 1000 subjects (n=1000), and once for big samples (n=10000).

Based on your answer from 6.a, compute the parameter values for each sampling distribution in R.

In [44]:
```
#poisson random sampling

#Question 6.b

set.seed(1)

# lambda = 1

# n= 10
y_rpois = rpois(10,lambda=1)
y_rpois
mean(y_rpois)

# n= 100
yy_rpois = rpois(100,lambda=1)
yy_rpois
mean(yy_rpois)

# n= 1000
yyy_rpois = rpois(1000,lambda=1)
yyy_rpois
mean(yyy_rpois)


# n= 10000
yyyy_rpois = rpois(10000,lambda=1)
yyyy_rpois
mean(yyyy_rpois)
```

```
# lambda = 5

# n= 10
y1_rpois = rpois(10,lambda=5)
y1_rpois
mean(y1_rpois)

# n= 100
yy1_rpois = rpois(100,lambda=5)
yy1_rpois
mean(yy1_rpois)

# n= 1000
yyy1_rpois = rpois(1000,lambda=5)
yyy1_rpois
mean(yyy1_rpois)


# n= 10000
yyyy1_rpois = rpois(10000,lambda=5)
yyyy1_rpois
mean(yyyy1_rpois)
```

```
# lambda = 20

# n= 10
y2_rpois = rpois(10,lambda=20)
y2_rpois
mean(y2_rpois)

# n= 100
yy2_rpois = rpois(100,lambda=20)
yy2_rpois
mean(yy2_rpois)

# n= 1000
yyy2_rpois = rpois(1000,lambda=20)
yyy2_rpois
mean(yyy2_rpois)


# n= 10000
yyyy2_rpois = rpois(10000,lambda=20)
yyyy2_rpois
mean(yyyy2_rpois)
```

1.0007

1.0027

9· 2· 6· 10· 2· 5· 5· 3· 10· 1

5.3

1· 7· 5· 5· 2· 5· 10· 4· 4· 6· 3· 9· 5· 4· 4· 6· 6· 5· 1· 7· 2· 12· 2· 6· 6· 1· 6· 2· 2· 7· 5·
2· 1· 3· 6· 4· 7· 2· 4· 4· 4· 2· 6· 6· 5· 8· 8· 5· 7· 2· 11· 4· 2· 2· 6· 5· 6· 4· 4· 6· 1· 12·
4· 3· 4· 2· 9· 3· 5· 4· 4· 5· 4· 2· 5· 9· 3· 6· 6· 4· 6· 8· 4· 6· 6· 3· 6· 4· 3· 9· 6· 3· 7·
5· 6· 2· 6· 2· 4· 3

4.8

4· 5· 4· 6· 5· 3· 4· 4· 6· 1· 7· 1· 5· 6· 9· 3· 2· 4· 8· 1· 8· 10· 2· 5· 2· 5· 1· 6· 7· 4· 5·
2· 2· 2· 4· 2· 4· 5· 5· 6· 4· 6· 4· 4· 10· 4· 7· 3· 9· 4· 4· 3· 0· 5· 6· 12· 3· 4· 9· 6· 0· 3·
6· 2· 8· 4· 6· 12· 2· 6· 5· 8· 5· 6· 10· 5· 5· 5· 4· 3· 4· 5· 5· 8· 5· 5· 9· 5· 6· 4· 4· 5· 4·
7· 8· 6· 2· 8· 4· 7· 6· 7· 5· 6· 1· 6· 6· 8· 4· 2· 9· 4· 9· 5· 10· 1· 3· 3· 5· 6· 2· 5· 4· 4·

Answers to 6.b

for lambda =1, the sample mean(which is the sample parameter)

when n is 10, sample mean = 1.1

when n is 100, sample mean = 1.01

when n is 1000, sample mean = 1.009

when n is 10000, sample mean = 1.0027

for lambda =5, the sample mean(which is the sample parameter)

when n is 10, sample mean = 5.3

when n is 100, sample mean = 4.8

when n is 1000, sample mean = 5.043

when n is 10000, sample mean = 4.9885

for lambda =20, the sample mean(which is the sample parameter)

when n is 10, sample mean = 23.6

when n is 100, sample mean = 19.93

when n is 1000, sample mean = 20.029

when n is 10000, sample mean = 20.0439

We can see here that, as sample size increases for each lambda, we can observe the sample mean converge to the population mean, which is lambda.

Question 6.c (5 Marks)

In this question, you are asked to experimentally justify the result in the CLT Theorem.

For different sample sizes of n = 10, 100 and 1000, use 50000 simulations (i.e. to approximate the infinite times we drew samples as mentioned before) to implement the sampling process.

From those 50000 sample means, compute the mean and standard deviation parameters (3 sample sizes and 3 λ rates, 9 pairs of parameters in total).

Discuss how the results reflect the CLT. Plot the results (mean and standard deviation separately) to demonstrate any effects you want to discuss.

In [45]:
```
#Question 6.c : experimentally justify the results in the CLT theorem

# 9 simulations for 9 pairs of parameters


set.seed(1)

n_sims = 50000

#lambda = 1, n = 10
lambda1 = 1
n1 = 10

sim1 = rpois(n1*n_sims, lambda1)
m1 = matrix(sim1, n_sims)

#sample means of 50000 simulations
sample_means1 = rowMeans(m1)


#mean of sample means
sm1.means = mean(sample_means1)
sm1.means
#sd of smaple means
sm1.std = sd(sample_means1)
sm1.std
# theoretically expected sd of disttribution of sample means, is actually close to that of
#our experiment
sm1.std.cltheorem = sqrt(lambda1/n1) # population sd




#lambda = 1, n = 100
lambda1 = 1
n2 = 100

sim2 = rpois(n2*n_sims, lambda1)
m2 = matrix(sim2, n_sims)

#sample means of 50000 simulations
sample_means2 = rowMeans(m2)

#mean of sample means
sm2.means = mean(sample_means2)
sm2.means
#sd of smaple means
sm2.std = sd(sample_means2)
sm2.std
# theoretical sd of dist
sm2.std.cltheorem = sqrt(lambda1/n2)




#lambda = 1, n = 1000
lambda1 = 1
n3 = 1000

sim3 = rpois(n3*n_sims, lambda1)
m3 = matrix(sim3, n_sims)

#sample means of 50000 simulations
sample_means3 = rowMeans(m3)

#mean of sample means
sm3.means = mean(sample_means3)
sm3.means
#sd of smaple means
sm3.std = sd(sample_means3)
sm3.std
# theoretical sd of dist
sm3.std.cltheorem = sqrt(lambda1/n3)
```

```
#lambda = 5, n = 10
lambda2 = 5
n1 = 10

sim4 = rpois(n1*n_sims, lambda2)
m4 = matrix(sim4, n_sims)

#sample means of 50000 simulations
sample_means4 = rowMeans(m4)

#mean of sample means
sm4.means = mean(sample_means4)
sm4.means
#sd of smaple means
sm4.std = sd(sample_means4)
sm4.std
# theoretical sd of dist
sm4.std.cltheorem = sqrt(lambda2/n1)
```

```
#lambda = 5, n = 100
lambda2 = 5
n2 = 100

sim5 = rpois(n2*n_sims, lambda2)
m5 = matrix(sim5, n_sims)

#sample means of 50000 simulations
sample_means5 = rowMeans(m5)

#mean of sample means
sm5.means = mean(sample_means5)
sm5.means
#sd of smaple means
sm5.std = sd(sample_means5)
sm5.std
# theoretical sd of dist
sm5.std.cltheorem = sqrt(lambda2/n2)
```

```
#lambda = 5, n = 1000
lambda2 = 5
n3 = 1000

sim6 = rpois(n3*n_sims, lambda2)
m6 = matrix(sim6, n_sims)

#sample means of 50000 simulations
sample_means6 = rowMeans(m6)

#mean of sample means
sm6.means = mean(sample_means6)
sm6.means
#sd of smaple means
sm6.std = sd(sample_means6)
sm6.std
# theoretical sd of dist
sm6.std.cltheorem = sqrt(lambda2/n3)
```

```
#lambda = 20, n = 10
lambda3 = 20
n1 = 10

sim7 = rpois(n1*n_sims, lambda3)
m7 = matrix(sim7, n_sims)

#sample means of 50000 simulations
sample_means7 = rowMeans(m7)
```

```
#mean of sample means
sm7.means = mean(sample_means7)
sm7.means
#sd of smaple means
sm7.std = sd(sample_means7)
sm7.std
# theoretical sd of dist
sm7.std.cltheorem = sqrt(lambda3/n1)
```

```
#lambda = 20, n = 100
lambda3 = 20
n2 = 100

sim8 = rpois(n2*n_sims, lambda3)
m8 = matrix(sim8, n_sims)

#sample means of 50000 simulations
sample_means8 = rowMeans(m8)

#mean of sample means
sm8.means = mean(sample_means8)
sm8.means
#sd of smaple means
sm8.std = sd(sample_means8)
sm8.std
# theoretical sd of dist
sm8.std.cltheorem = sqrt(lambda3/n2)
```

```
#lambda = 20, n = 1000
lambda3 = 20
n3 = 1000

sim9 = rpois(n3*n_sims, lambda3)
m9 = matrix(sim9, n_sims)

#sample means of 50000 simulations
sample_means9 = rowMeans(m9)

#mean of sample means
sm9.means = mean(sample_means9)
sm9.means
#sd of smaple means
sm9.std = sd(sample_means9)
sm9.std
# theoretical sd of dist
sm9.std.cltheorem = sqrt(lambda3/n3)
```

```
0.999366

0.316047141673383

0.9995202

0.0995717735240347

0.99991626

0.0316929243908089

5.006264

0.709301645620881

5.0019214

0.223642872778747

5.00041164

0.0709613070810597

19.98883
```

1.41967818213242

20.0018044

0.445767152567039

20.00016552

0.141983734106216

for lambda = 1, the sample mean(which is the sample parameter) and sample standard deviation

when n is 10, sample mean = 0.999366, and sample standard deviation = 0.316047141673383

when n is 100, sample mean = 0.9995202, and sample standard deviation = 0.0995717735240347

when n is 1000, sample mean = 0.99991626, and sample standard deviation = 0.0316929243908089

for lambda = 5, the sample mean(which is the sample parameter) and sample standard deviation

when n is 10, sample mean = 5.006264, and sample standard deviation = 0.709301645620881

when n is 100, sample mean = 5.0019214, and sample standard deviation = 0.223642872778747

when n is 1000, sample mean = 5.00041164, and sample standard deviation = 0.0709613070810597

for lambda = 20, the sample mean(which is the sample parameter) and sample standard deviation

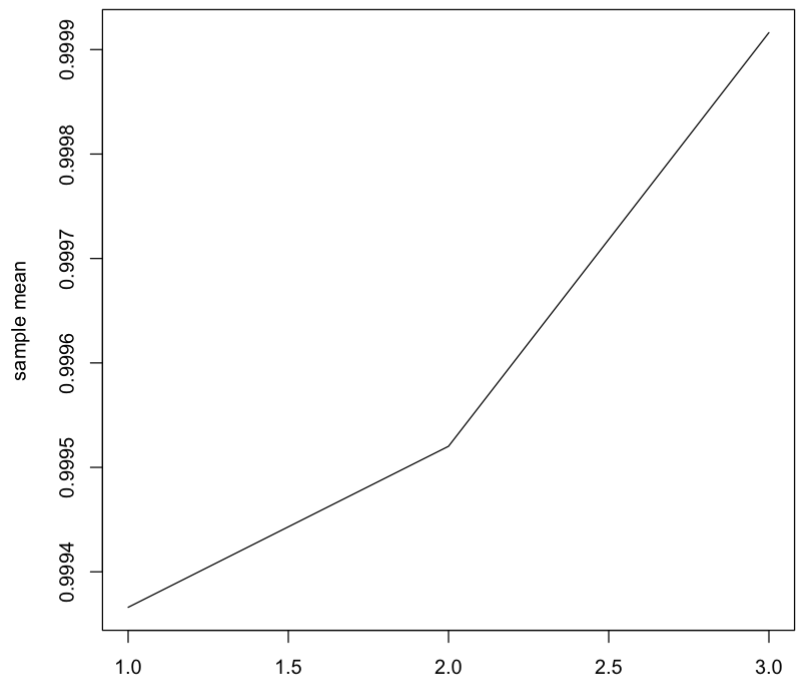
when n is 10, sample mean = 19.98883, and sample standard deviation = 1.41967818213242

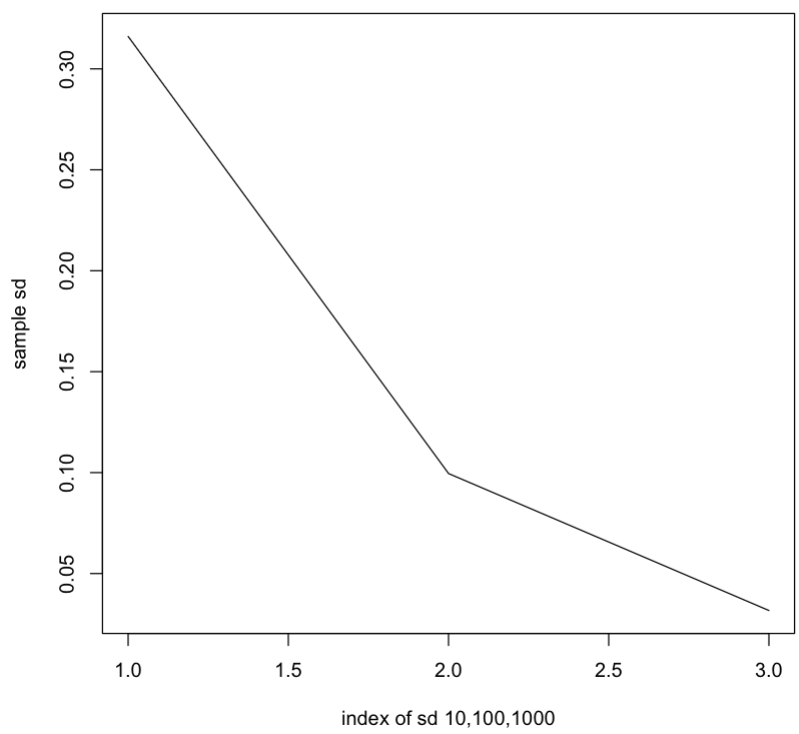
when n is 100, sample mean = 20.0018044, and sample standard deviation = 0.445767152567039

when n is 1000, sample mean = 20.00016552, and sample standard deviation = 0.141983734106216

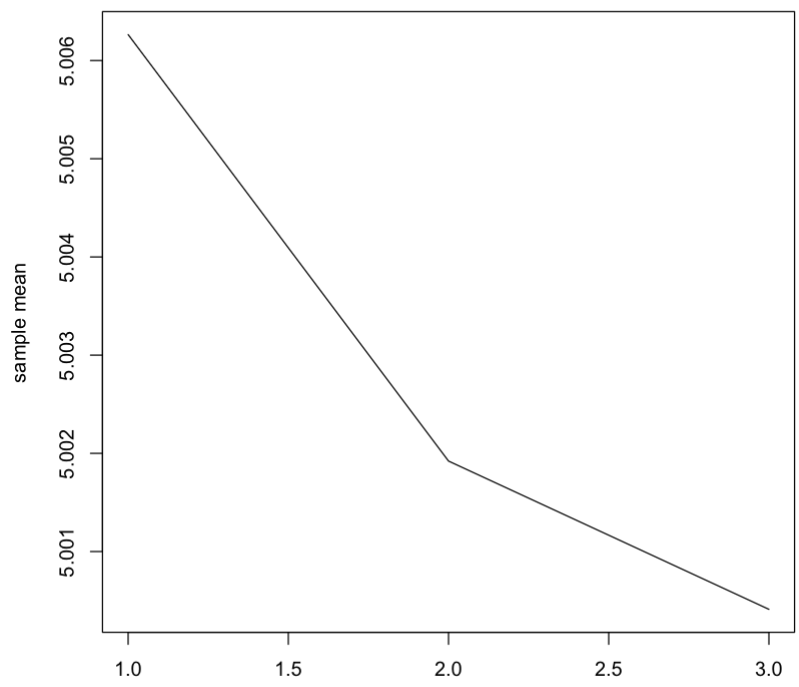
We can see that for each pair the mean of sample means are very close to the lambda value(which is the mean of a poisson dist). we can also observe that as sample size increases, the estimate of the mean is better. Interesting to note that sd decreases as sample size gets larger.

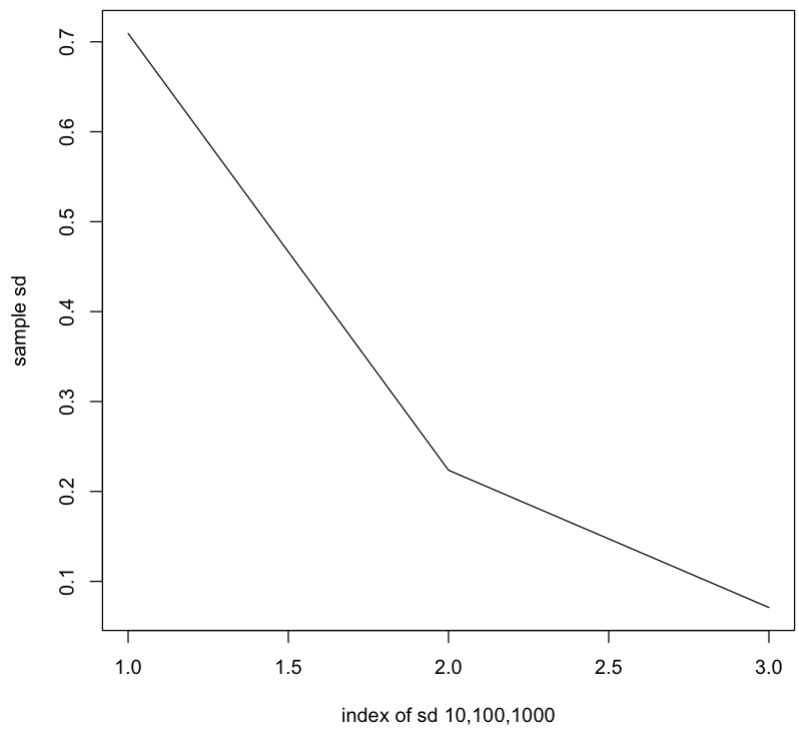
In [46]:
```
# plot for parameters when lambda = 1
# plots of mean and sd

lambda1_plot_mean = c(sm1.means,sm2.means,sm3.means)
plot(lambda1_plot_mean,type='l', xlab = 'index of sample means 10,100,1000', ylab = 'sample mean')

lambda1_plot_sd = c(sm1.std,sm2.std,sm3.std)
plot(lambda1_plot_sd,type='l', xlab = 'index of sd 10,100,1000', ylab = 'sample sd')
```



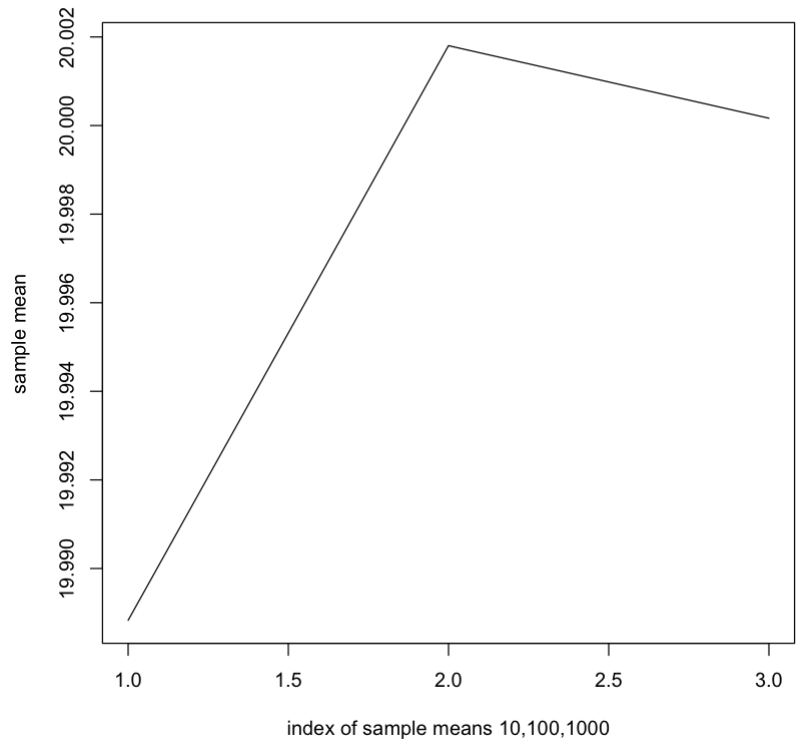


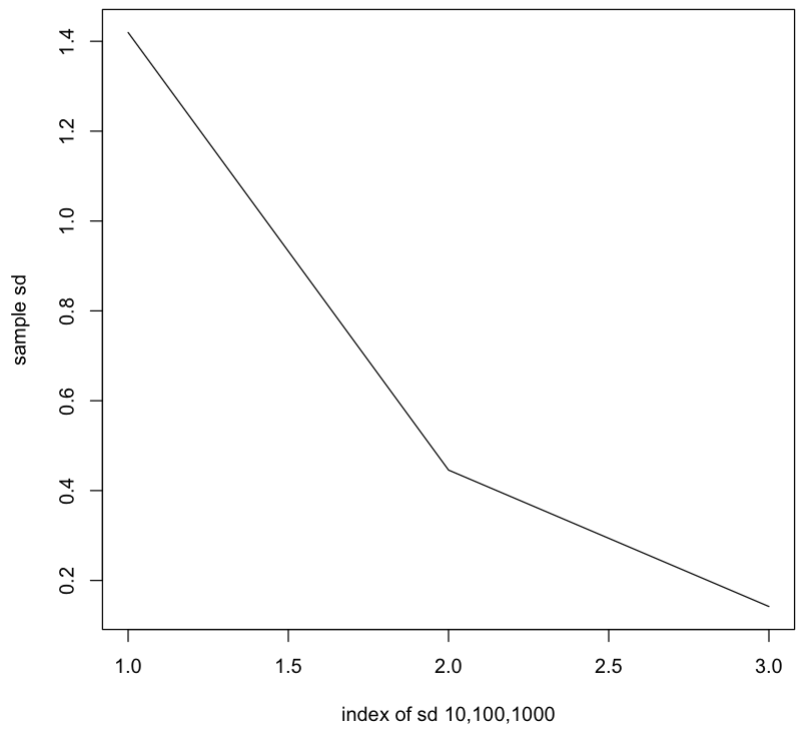
In [47]:
```
# plot for parameters when lambda = 5
# plots of mean and sd

lambda2_plot_mean = c(sm4.means,sm5.means,sm6.means)
plot(lambda2_plot_mean,type='l', xlab = 'index of sample means 10,100,1000', ylab = 'sample mean')

lambda2_plot_sd = c(sm4.std,sm5.std,sm6.std)
plot(lambda2_plot_sd,type='l', xlab = 'index of sd 10,100,1000', ylab = 'sample sd')
```





In [48]:
```
# plot for parameters when lambda = 20
# plots of mean and sd

lambda3_plot_mean = c(sm7.means,sm8.means,sm9.means)
plot(lambda3_plot_mean,type='l', xlab = 'index of sample means 10,100,1000', ylab = 'sample mean')

lambda3_plot_sd = c(sm7.std,sm8.std,sm9.std)
plot(lambda3_plot_sd,type='l', xlab = 'index of sd 10,100,1000', ylab = 'sample sd')
```





We can observe from the above plots that: the plots of means dont tell us much. but for the plots of sample standard deviations, we can see that as sample size increases, the sample standard deviations decrease. Also sample standard deviation is approximately close to the standard deviation of the population. Also interesting to note that, the sample standard deviation differences for each lambda is proportional.

Question 6.d (5 Marks)

When rate λ_1 = 1 and λ_2 = 5 and sample size n is 10 or 100, obtain the z scores of the sample means (from 50000 simulations). Plot their distributions in a histogram with the theoretical Gaussian curve overlaid.

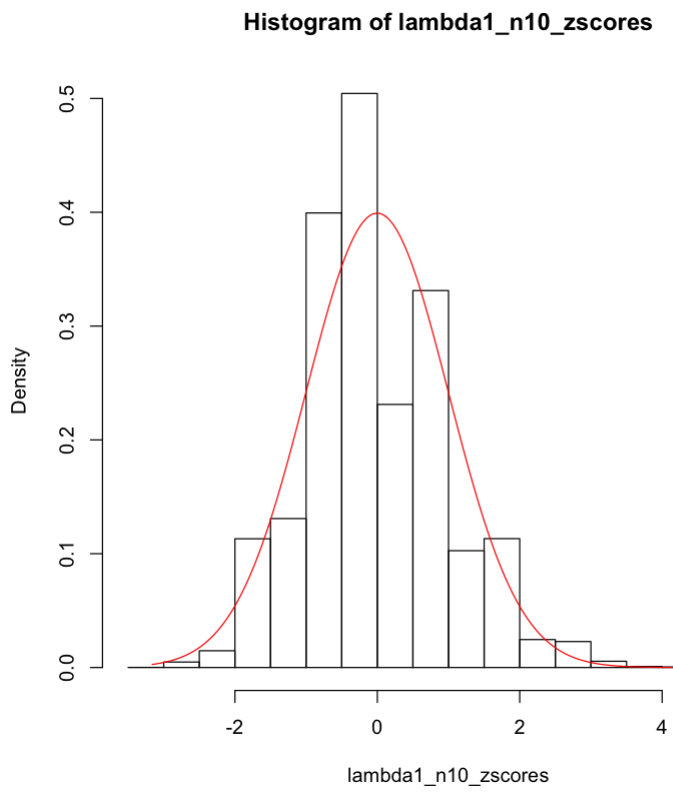
Note that for sample size 100, the plots overlay very nicely. But what happens with sample size 10? Explain the differences between the four plots.

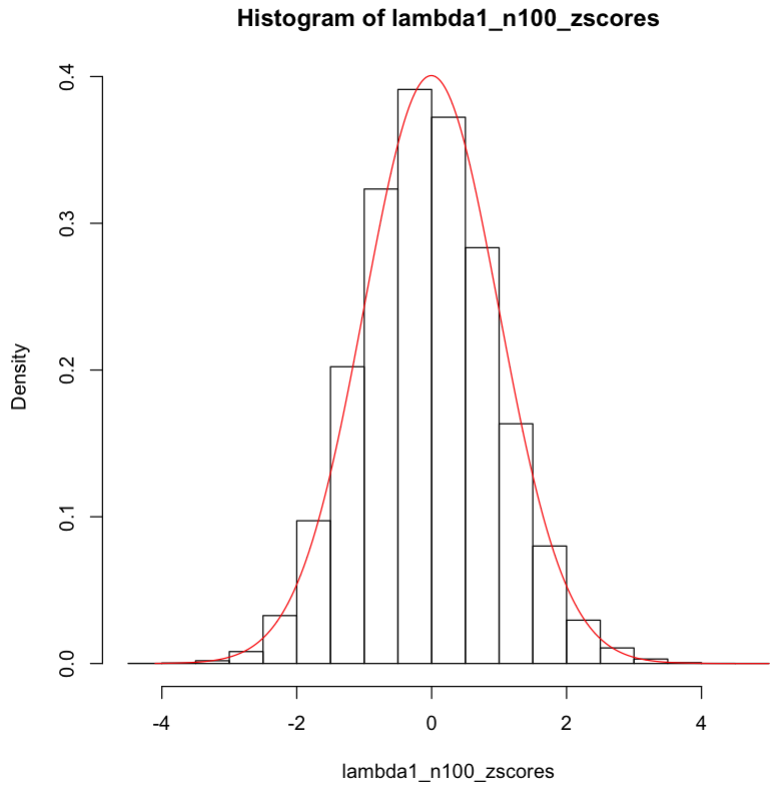
For each simulation: the z score of the mean can be calculated as:

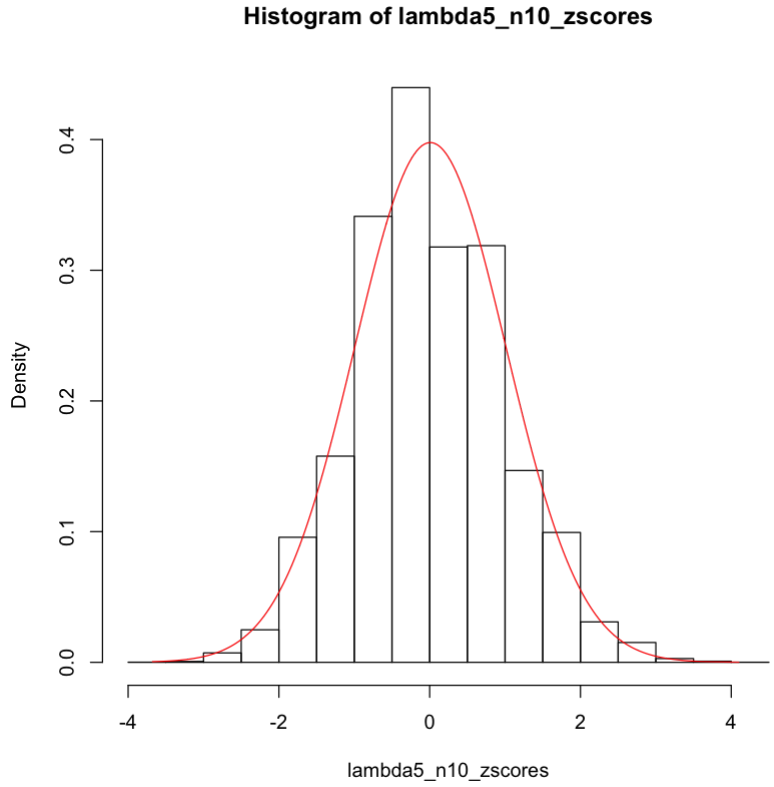
$$\frac{(\bar{X} - \mu)}{\sigma/\sqrt{n}}$$

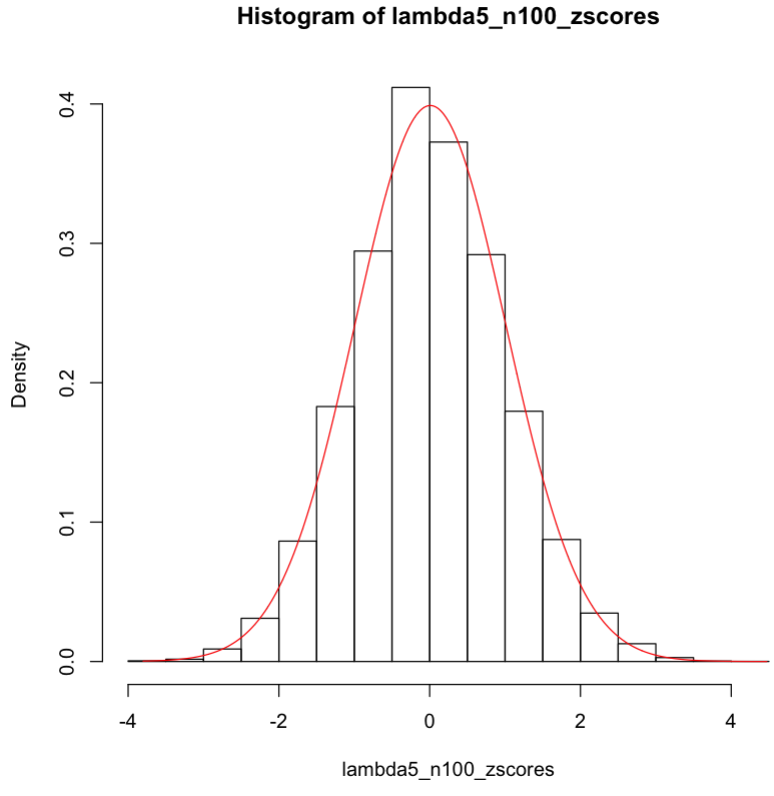
where $\bar{X}$ is the mean of the sample, μ is the population mean and σ is the population standard deviation.

In [49]:
```
#question 6d.


#creating zscore histograms of density

#sample_means1, where lambda = 1, n = 10
lambda1_n10_zscores = lapply(sample_means1,function(sample_means1) ((sample_means1-lambda1)/(sqrt(lambda1)/sqrt(
lambda1_n10_zscores <- unlist(lambda1_n10_zscores, use.names = FALSE)
hist(lambda1_n10_zscores,freq = F)
points(seq(min(lambda1_n10_zscores), max(lambda1_n10_zscores),length.out = 500),
       dnorm(seq(min(lambda1_n10_zscores), max(lambda1_n10_zscores),length.out = 500),
             mean(lambda1_n10_zscores), sd(lambda1_n10_zscores)),type='l', col="red")



#sample_means2, where lambda = 1, n = 100
lambda1_n100_zscores = lapply(sample_means2,function(sample_means2) ((sample_means2-lambda1)/(sqrt(lambda1)/sqrt
lambda1_n100_zscores <- unlist(lambda1_n100_zscores, use.names = FALSE)
hist(lambda1_n100_zscores,freq=F)
points(seq(min(lambda1_n100_zscores), max(lambda1_n100_zscores),length.out = 500),
       dnorm(seq(min(lambda1_n100_zscores), max(lambda1_n100_zscores),length.out = 500),
             mean(lambda1_n100_zscores), sd(lambda1_n100_zscores)),type='l', col="red")

#sample_means4, where lambda = 5, n = 10
lambda5_n10_zscores = lapply(sample_means4,function(sample_means4) ((sample_means4-lambda2)/(sqrt(lambda2)/sqrt(
lambda5_n10_zscores <- unlist(lambda5_n10_zscores, use.names = FALSE)
hist(lambda5_n10_zscores,freq=F)
points(seq(min(lambda5_n10_zscores), max(lambda5_n10_zscores),length.out = 500),
       dnorm(seq(min(lambda5_n10_zscores), max(lambda5_n10_zscores),length.out = 500),
             mean(lambda5_n10_zscores), sd(lambda5_n10_zscores)),type='l', col="red")

#sample_means5, where lambda = 5, n = 100
lambda5_n100_zscores = lapply(sample_means5,function(sample_means5) ((sample_means5-lambda2)/(sqrt(lambda2)/sqrt
lambda5_n100_zscores <- unlist(lambda5_n100_zscores, use.names = FALSE)
hist(lambda5_n100_zscores,freq=F)
points(seq(min(lambda5_n100_zscores), max(lambda5_n100_zscores),length.out = 500),
       dnorm(seq(min(lambda5_n100_zscores), max(lambda5_n100_zscores),length.out = 500),
             mean(lambda5_n100_zscores), sd(lambda5_n100_zscores)),type='l', col="red")
```









Answers 6.d: describing observation from plots

For the sample size of n=10 (for each lambda equal to 1 and 5) the gaussian curve does not overlay nicely. This is because, at smaller sample sizes the histogram still has slight right skewness but the gaussian overlay is perfectly normal(proportional). Thus they do not match properly. On the otherhand for n=100(for each lambda equal to 1 and 5) the poisson distribution starts to normalise and the gaussian overlay can be observed as almost perfect overlay. This is the essense of central limit theorem, where even though the underlying distribution for our simulation was a poisson distribution, for larger sample sizes, it converges into a gaussian distribution.

In [ ]: