

Machine Learning

Home Work 01

aqeel labash

13 February 2016

First Question

First(a)

To replace numerical codes to labels I used factor. Factor method replace the value that match with corresponding label (original values , values to be matched, Corresponding labels).(**Note:** I initiated all the vectors on the fly)

```
1 #Convert Landmass
2 data$landmass <- factor(data$landmass,c(1:6),c("N.America", "S.America", "Europe", "Africa",
3 "Asia", "Oceania"))
4 #Convert zone
5 data$zone <- factor(data$zone,c(1:4),c('NE', 'SE', 'SW', 'NW'))
6 unique(data$language)
7 #Concert Language
8 data$language<- factor(data$language,c(1:10),c('English', 'Spanish', 'French', 'German', '
9 Slavic', 'Other Indo-European', 'Chinese', 'Arabic', 'Japanese/Turkish/Finnish/Magyar', '
10 Others'))
11 #Convert Religion
12 unique(data$religion)
13 data$religion = factor(data$religion,c(0:7),c('Catholic', 'Other Christian', 'Muslim', '
14 Buddhist',
15 'Hindu', 'Ethnic', 'Marxist', 'Others'))
```

Figure 1 shows a comparison between religion in number of countries.

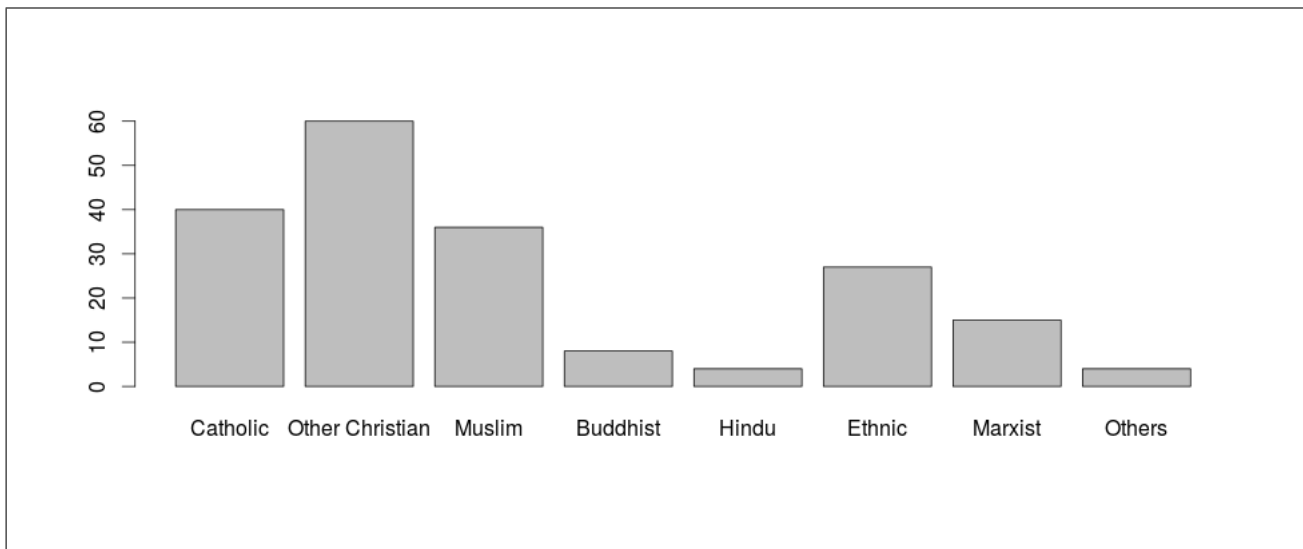


Figure 1: Countries Religion

Figure 2 shows a comparison between landmasses in number of countries.(how many countries in each landmass)

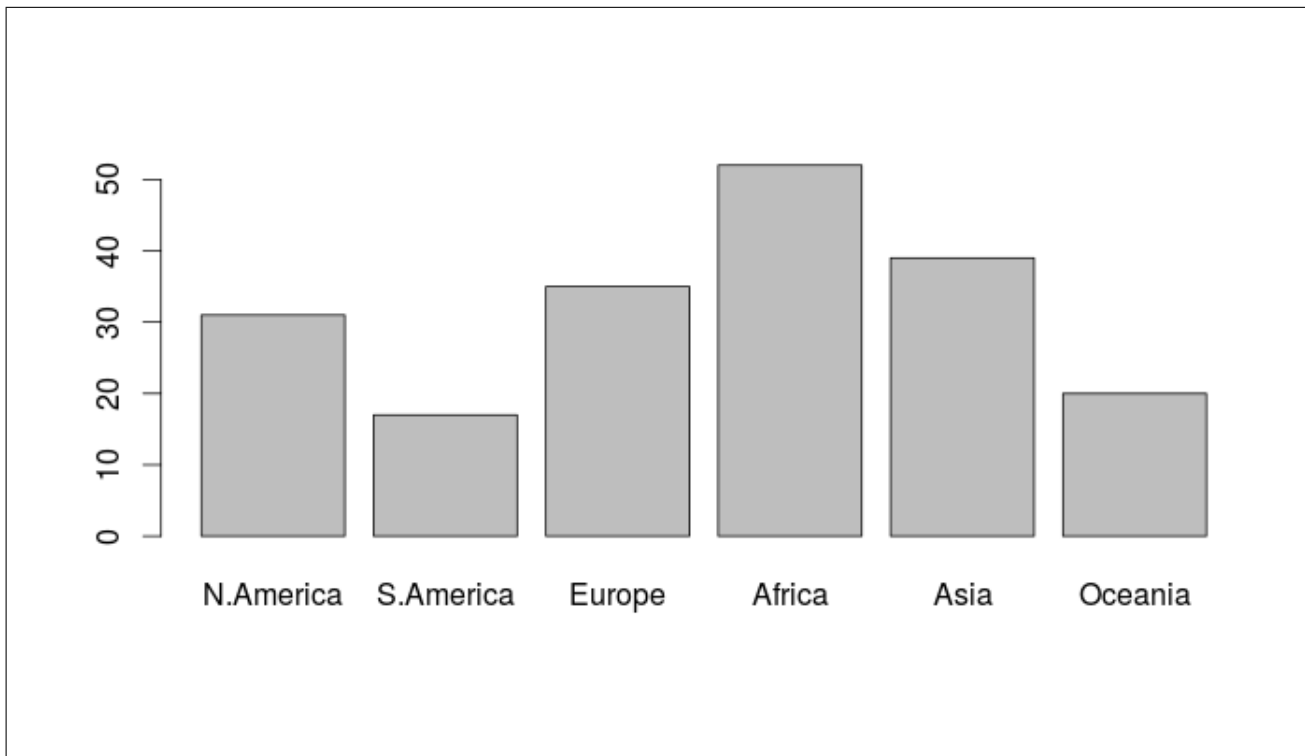


Figure 2: Countries landmass

Figure 3 shows a comparison between languages in number of countries.(how many countries has that language as official)

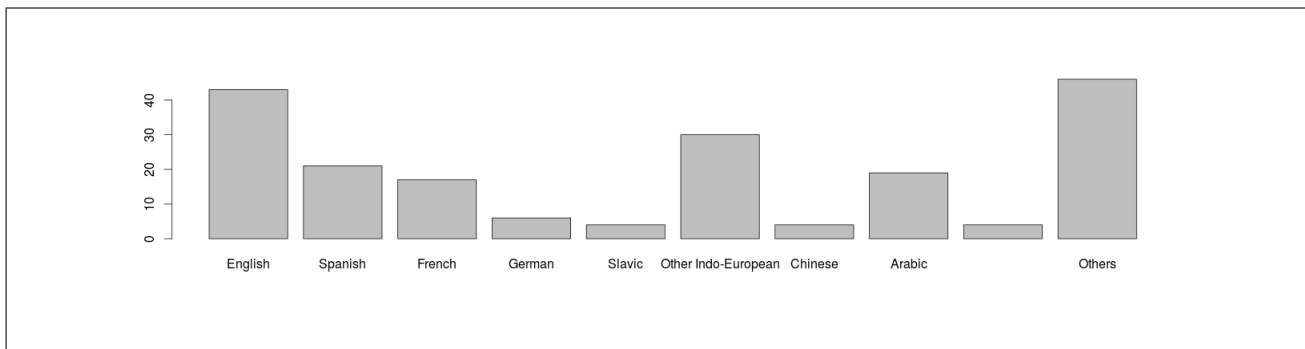


Figure 3: Countries landmass

The previous plots where done by the following code where the data converted to a table and then the plot has been done.

```

1 #religion barplot
2 barplot(table(data$religion))
3 #landmass barplot
4 barplot(table(data$landmass))
5 #language barplot
6 barplot(table(data$language))
7

```

First(b)

(I interpreted this task to show how religion & landmass effect the area and population) For this task first I filtered the data to Africa and Europe only.Then I dropped the levels so only Africa and Europe will show up in the plot

```

1 #First Isolate the data I want to work on for this question.
2 importadata = subset (data,landmass=="Europe" | landmass == "Africa")
3
4 #drop levels of landmass (landmass other than(Europe , Africa) won't show up in the plot)
5 importadata$landmass <- droplevels(importadata$landmass)
6 #plot area vs landmass

```

as shown in figure 4 we can notice that Africa countries has more area than Europe countries. As well we can notice in Africa we have two exceptions (Sudan and Algeria) (**Soviet Union in dataset still exist and it's in Asia**)

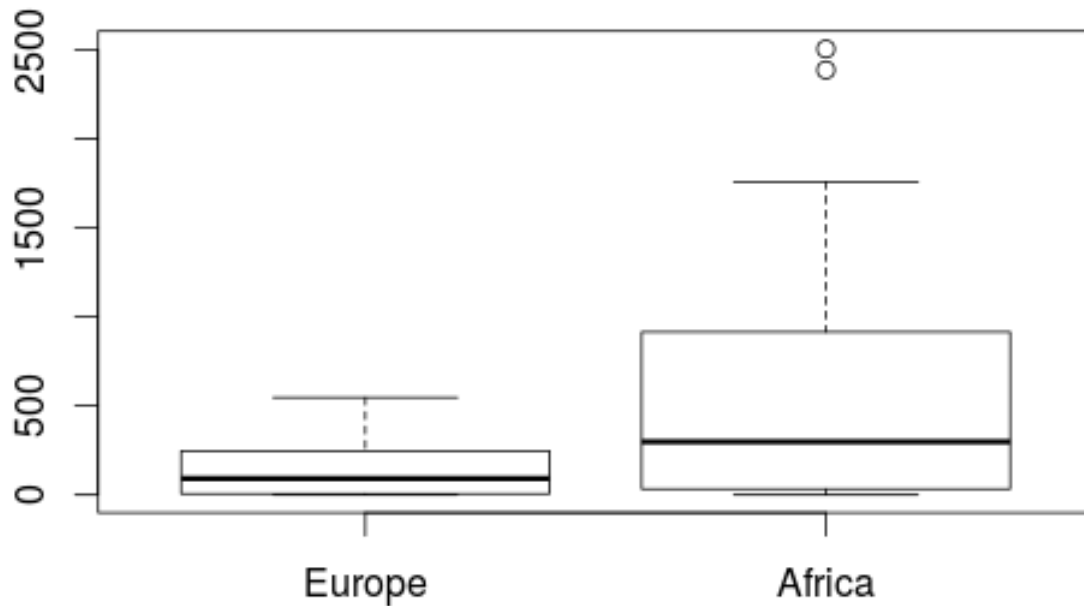


Figure 4: Countries area corresponding to landmass(Europe and Africa)

Figure 5 shows that countries in Europe has more population than African countries.

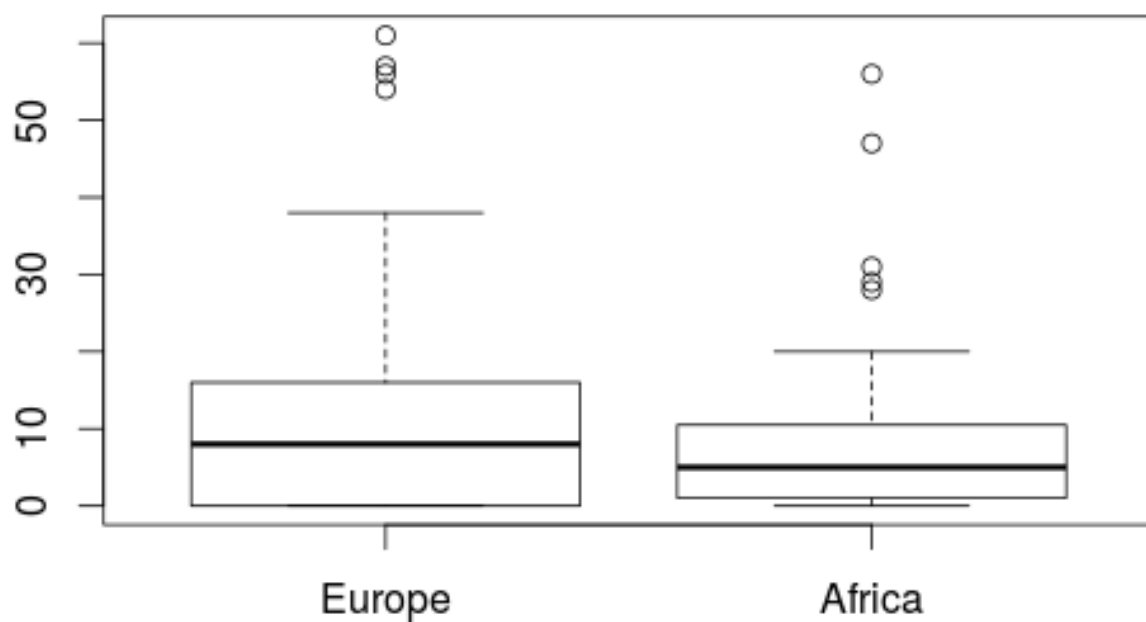


Figure 5: Countries population corresponding to landmass(Europe and Africa)

Now to see the area and population dependency on religion

In Figure 6 we clearly notice that Hindu countries has wide area. But in the same time it's median is very low which mean that countries above the median is very distributed. Marxist has high median which mean that countries above the median is not very distributed.

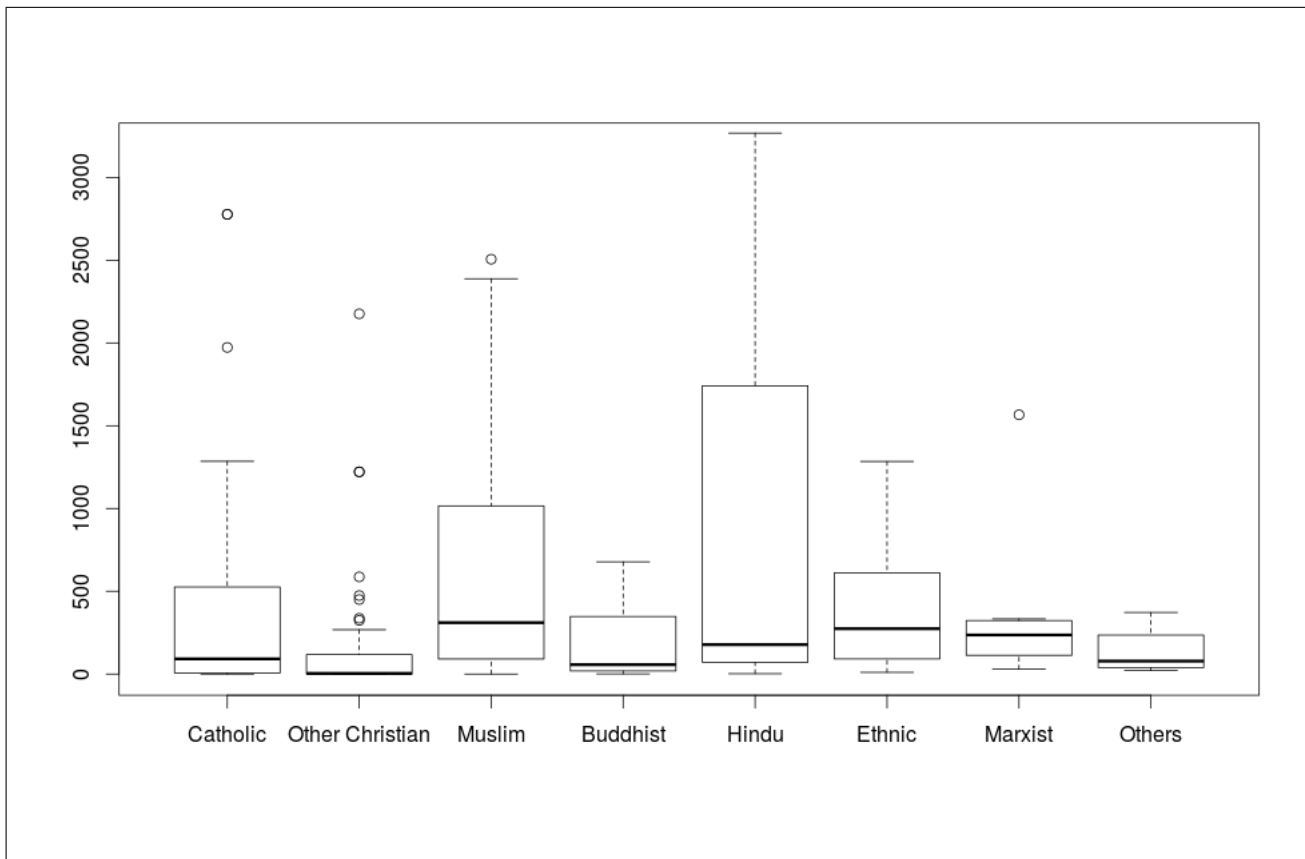


Figure 6: Countries area corresponding to religion

Just to clarify the previous figure I calculated number of Hindu & Marxist & Muslim countries and I found that there is 4 Hindu countries which explain why it's was very distributed. There is 36 Muslim countries and we can tell that Muslim countries has high probability to have more area than (Other Christian) countries.

```

1 #Number of Hindu Countries
2 length(subset(data, religion=='Hindu')$landmass)
3 #Number of Marxist Countries
4 length(subset(data, religion=='Marxist')$landmass)
5 #Number of Muslim Countries
6 length(subset(data, religion=='Muslim')$landmass)
7

```

In figure 7 we can see boxplot for Countries population corresponding to religion but to make it more clear I'll minimize the range in figure 8.

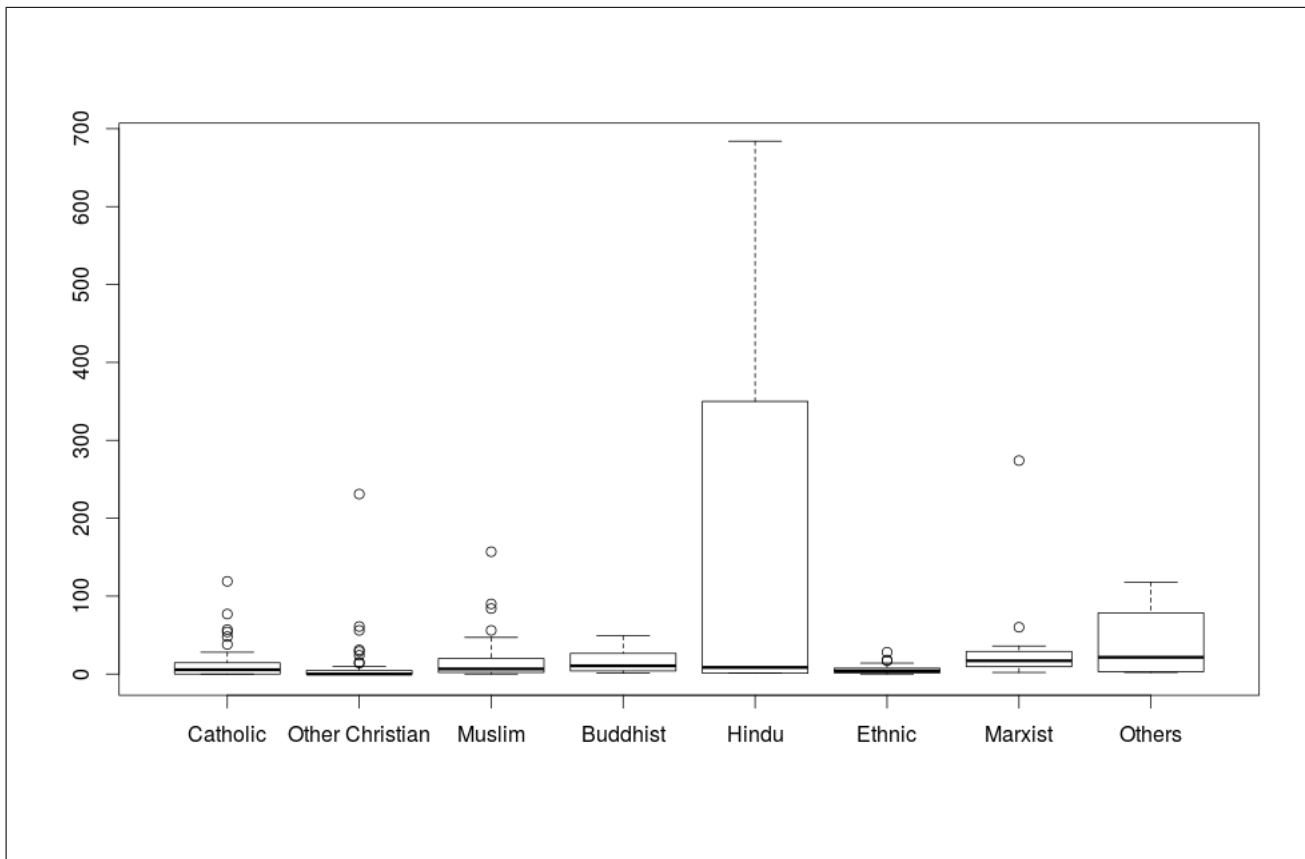


Figure 7: Countries population corresponding to religion

In figure 8 we can see that (Hindu,Others) countries have high population and Hindu countries has high probability to have high population over other religion.

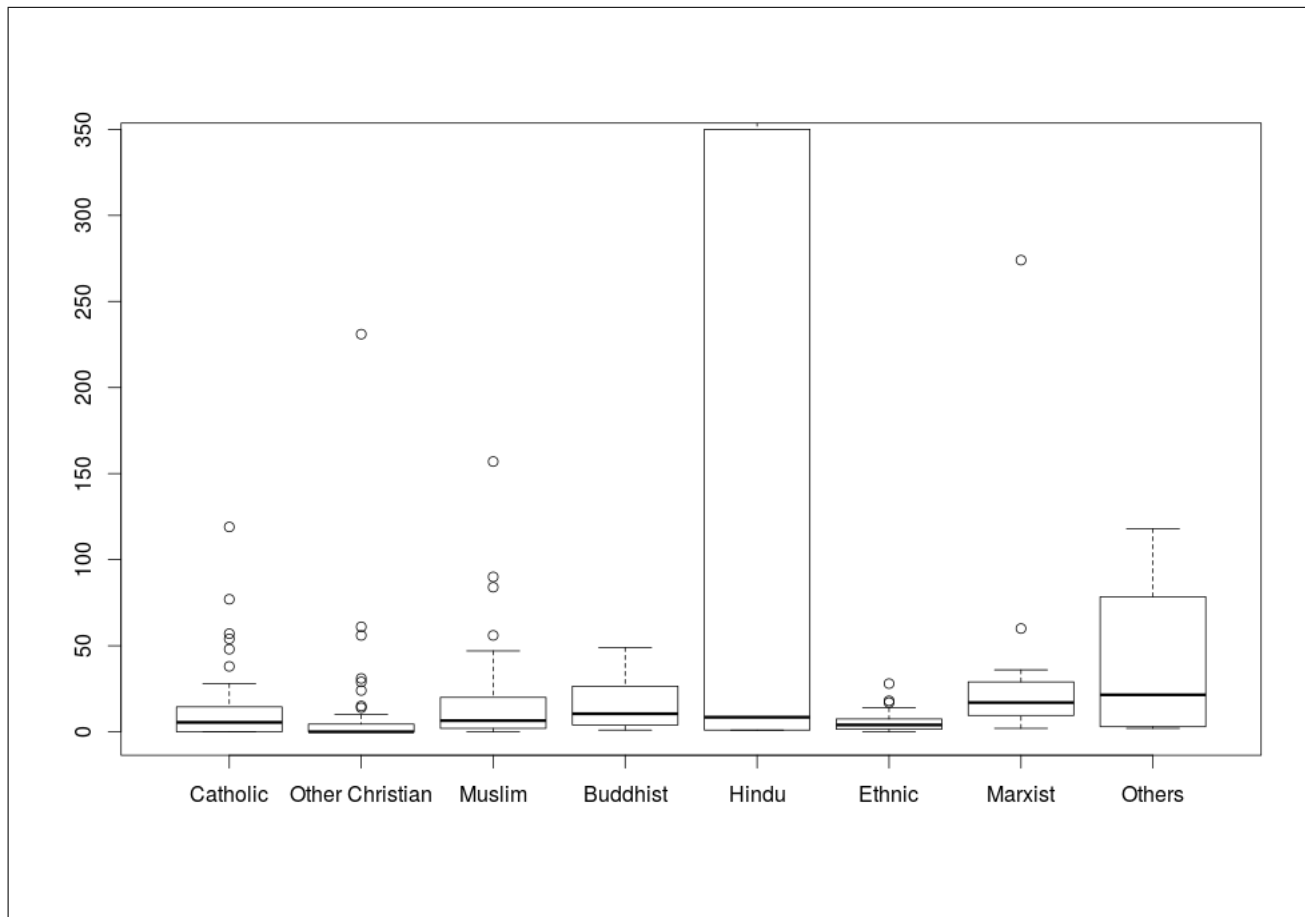


Figure 8: Countries population corresponding to Religion (mimized)

The Previous plots were generated by this code.

```

1 #plot area vs landmass
2 boxplot(area ~ landmass, data=importadata)
3
4 #plot popluation vs landmass
5 boxplot(population ~ landmass, data=importadata)
6
7 #plot area vs religion
8 boxplot(area ~ religion, data, ylim=c(0,3200))
9
10 #plot population vs religion
11 boxplot(population ~ religion, data, ylim=c(0,680))
12
13 #plot population vs religion (minimize scale)
14 boxplot(population ~ religion, data, ylim=c(0,340))
15

```

Note:the domain (0,3200) and (0,680) were selected as less as possible to have a better view over the plots

First(c)

To compute quantiles I created vector contain values from 0.1 to 0.9 (because the question requested from 10% to 90%). After that computation were done for area, population and density ($\text{density} = \frac{\text{population} * 1000000}{\text{area} * 1000}$) by the following code:

```

1 #Basic Variables
2 qlvls = c(1:9)/10
3 dataEurope<-subset(data, landmass=="Europe")
4 dataAfrica<-subset(data, landmass=="Africa")
5 #quantile area
6 quant.Europe.area<-quantile(dataEurope$area, qlvls)
7 quant.Africa.area<-quantile(dataAfrica$area, qlvls)
8
9 #quantile population
10 quant.Europe.population<-quantile(dataEurope$population, qlvls)
11 quant.Africa.population<-quantile(dataAfrica$population, qlvls)

```

```

12 #quantile density
13 EuropeDensity <- (dataEurope$population*1000000)/(dataEurope$area*1000)
14 AfricaDensity <- (dataAfrica$population*1000000)/(dataAfrica$area*1000)
15
16

```

Figure 9 shows that countries in Africa larger and distributed over all quantiles in the other hand Europe not distributed over all quantiles.

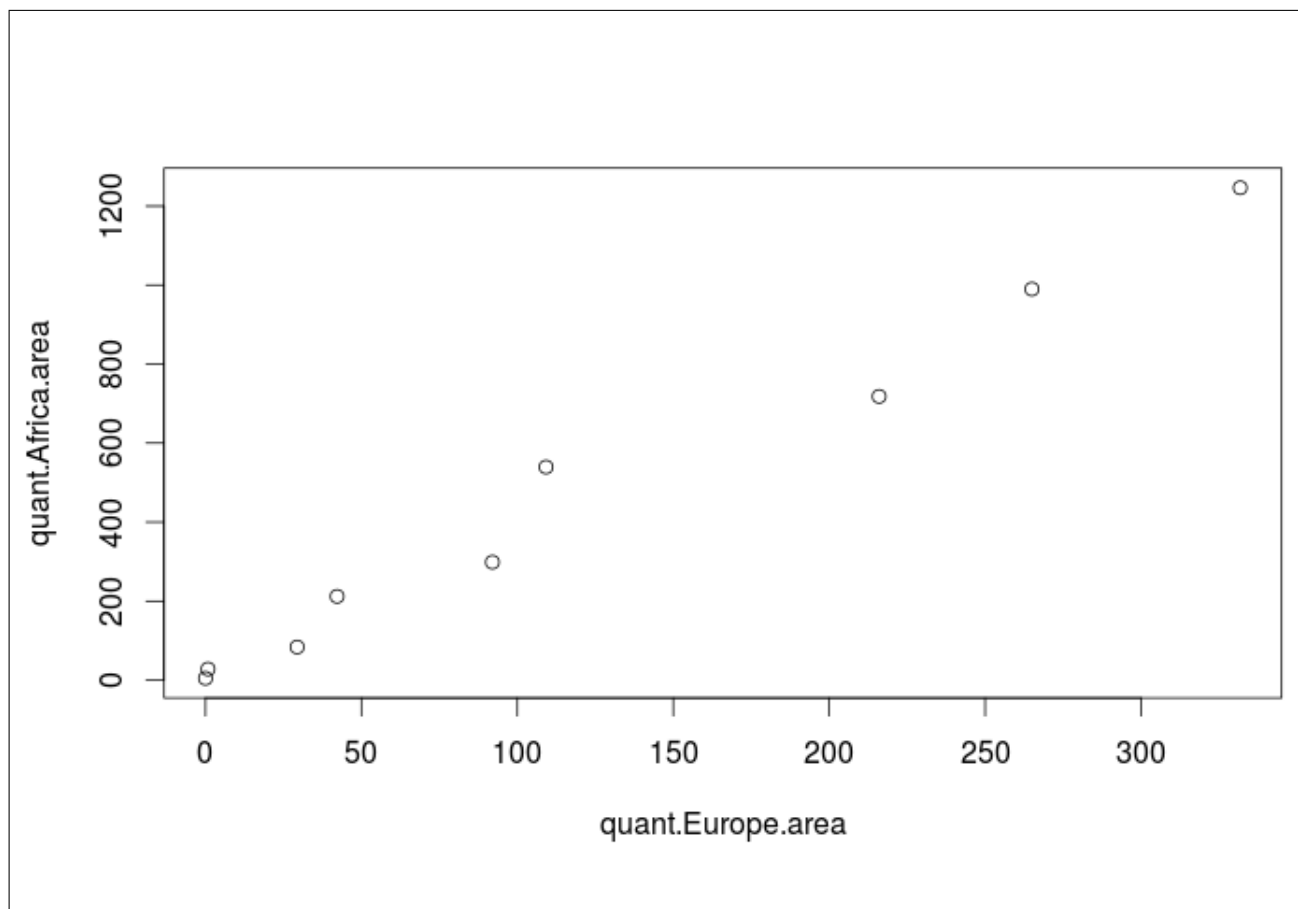


Figure 9: Countries area in Europe vs Africa

In figure 9 we can see that Africa countries has more areas than Europe ones.

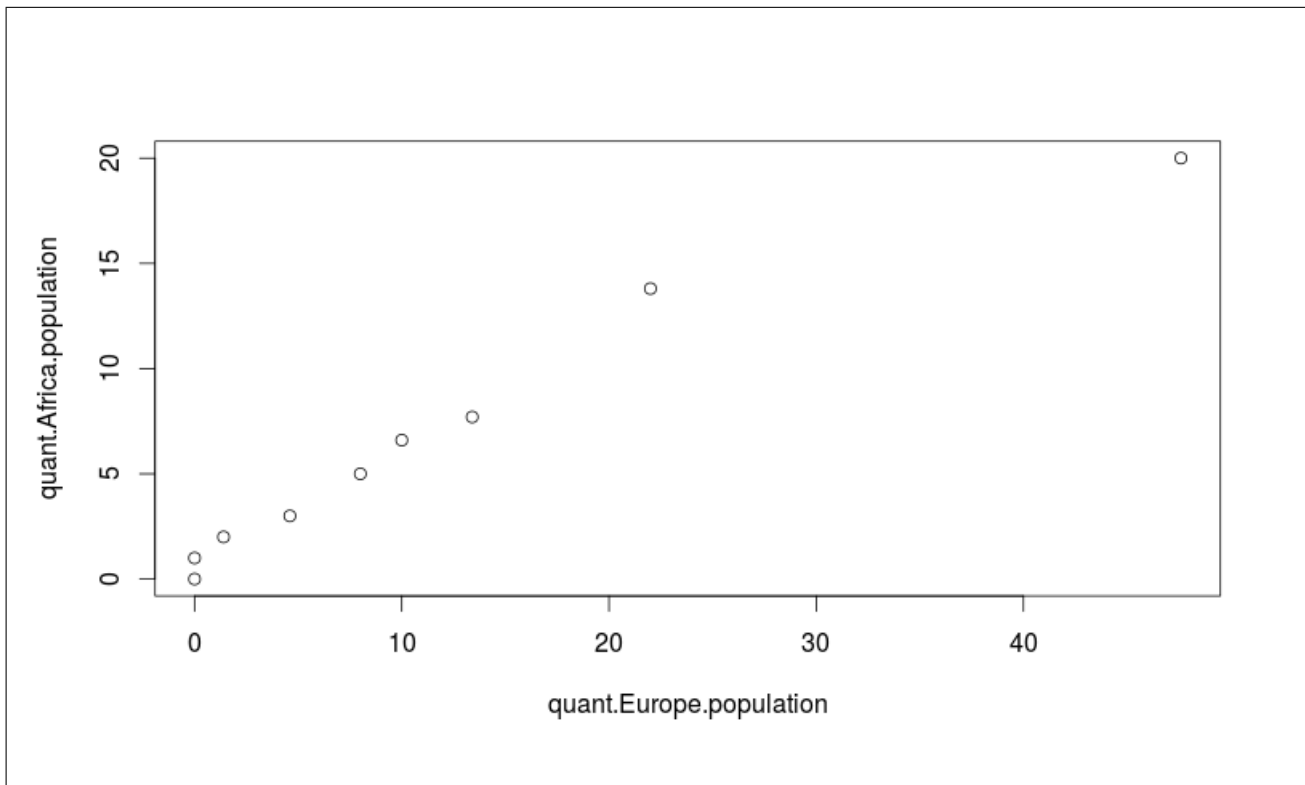


Figure 10: Countries population in Europe vs Africa

In figure 10 we can observe how Europe countries has more population than African countries.

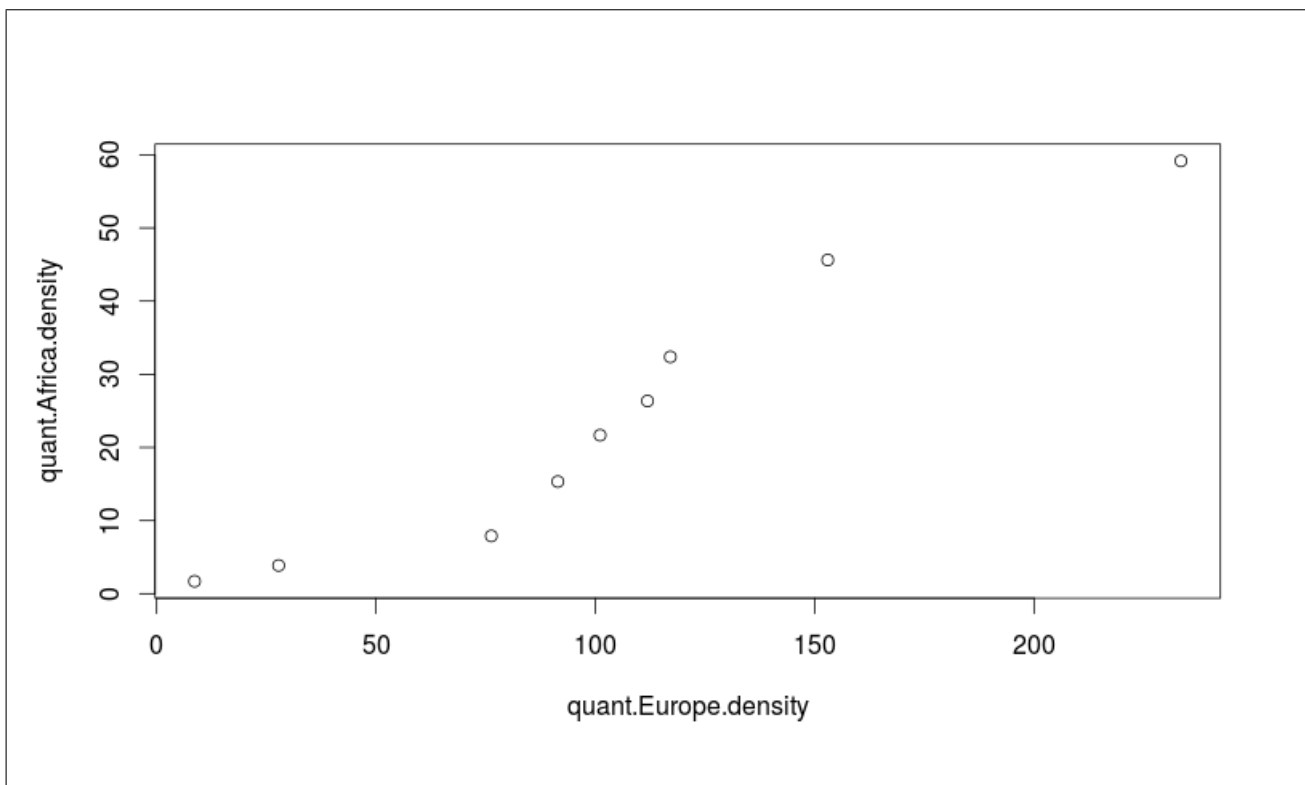


Figure 11: Countries density in Europe vs Africa

In figure 11 it's clearly that Europe has higher density and that's due Africa has larger area but less population as we saw in figure 9&10

First(d)

To get the cover we get number of rows for (Marxist, Muslim , English).

Support we get the number of rows for the condition and the result (Marxist&red , Muslim&green , English&saltires>0).

Relative Support : Support to total data.

Confidence: Support / Cover. and here is the code used for it:

```
1 #Cover
2 Marxist.Cover = nrow(subset(data, religion=="Marxist"))
3 Muslims.Cover = nrow(subset(data, religion=="Muslim"))
4 English.Cover = nrow(subset(data, language=="English"))
5
6 #Support
7 Marxist.Support = nrow(subset(data, religion=="Marxist" & red=="1"))
8 Muslim.Support = nrow(subset(data, religion=="Muslim" & green=="1"))
9 English.Support = nrow(subset(data, language=="English" & saltires > 0))
10
11 #Relative Support
12 Marxist.RelativeSupport = Marxist.Support/nrow(data)*100
13 Muslim.RelativeSupport = Muslim.Support/nrow(data)*100
14 English.RelativeSupport = English.Support/nrow(data)*100
15
16 #Confidence
17 Marxist.Confidence = Marxist.Support/Marxist.Cover
18 Muslims.Confidence = Muslim.Support/Muslims.Cover
19 English.Confidence = English.Support/English.Cover
20
```

The result was :

Marxist Cover : 15 , Support : 15 , Relative Support : 7.731959, Confidence : 1.

Muslims Cover : 35 , Support : 26 , Relative Support : 13.40206, Confidence : 0.72

English Cover : 43, Support : 15 , Relative Support : 7.731959, Confidence: 0.3488372

First(e)

As I understood the question here I need to find the best rules for 3 cases :saltires , crosses , (Sun or star)

Saltires:Firstly I analyzed the data and I noticed that English & Other Christian were the most common between all saltires.And here is the code:

```
1 #analyzing saltires data
2 testdata <- subset(data, saltires > 0)
3 #trying rules
4 saltires.English.Support <- nrow(subset(data, language=="English" & saltires > 0))
5 saltires.English.Confidence <- saltires.English.Support / nrow(subset(data, language=="English"
6 ))
7 saltires.OtherChristian.Support <- nrow(subset(data, religion == "Other Christian" & saltires > 0))
8 saltires.OtherChristian.Confidence <- saltires.OtherChristian.Support / nrow(subset(data,
9 religion=="Other Christian"))
10 saltires.Zone.Support <- nrow(subset(data, zone=="NW" & saltires > 0))
11 saltires.Zone.Confidence <- saltires.Zone.Support / nrow(subset(data, zone=="NW"))
```

Using English language as the rule I got :Support : 15, Confidence:0.348837.

Using religion "Other Christian" as the rule I got: Support :16 , Confidence : 0.26.

Using zone "NW" as the rule I got : Support : 7 , Confidence: 0.12.

Obviously English rule got the highest confidence after that the religion and that's as I looked into it it's related to Scotland and saint Andrews.

Crosses:I did as before first analyzed the data were it was the same English & Other Christian & blue as bottom right color of the flag were most common shared characteristics between the data and here is the code :

```
1 #analyzing crosses
2 testdata <- subset(data, crosses > 0)
3
4 #trying rules
5 crosses.OtherChristian.Support <- nrow(subset(data, religion=="Other Christian" & crosses > 0))
6 crosses.OtherChristian.Confidence <- crosses.OtherChristian.Support / nrow(subset(data, religion
7 == "Other Christian"))
8 crosses.English.Support <- nrow(subset(data, language=="English" & crosses > 0))
```

```

9 crosses . English . Confidence<-crosses . OtherChristian . Support/nrow(subset(data , language=="
  English" ))
10
11 crosses . blue . Support<-nrow(subset(data , botright==" blue"&crosses >0))
12 crosses . blue . Confidence<-crosses . blue . Support/nrow(subset(data , botright==" blue" ))
13
14 crosses . zone . Support <-nrow(subset(data , zone=="NE"&crosses >0))
15 crosses . zone . Confidence<-crosses . zone . Support/nrow(subset(data , zone=="NE" ))
16

```

The results were as following :

Using English language as rule : Support : 14 , Confidence: 0.55813.

Using "Other Christian" religion as rule : Support: 24 , Confidence:0.4.

Using blue color for bottom right as rule : Support 17, Confidence : 0.36

Using NE for zone as rule : Support 8, Confidence : 0.87.

Actually here I believe the cross is related to the history of NE zone. So if I have to pick a rule I would pick it event if this rule have small support but with high confidence it's better to go with I believe.

SunsorStars: This was hard to analyze and I used a new way to tackle it. First the code then the result.

```

1 #analyzing Sunsorstars
2 testdata<-subset(data , sunstars >0)
3 #Check By Language
4 table(testdata$language)
5 table(testdata$language)/table(data$language)
6 #Check By landmass
7 table(testdata$landmass)
8 table(testdata$landmass)/table(data$landmass)
9 #Check By religion
10 table(testdata$religion)
11 table(testdata$religion)/table(data$religion)
12 #Check By Zone
13 table(testdata$zone)
14 table(testdata$zone)/table(data$zone)
15

```

After executing the above code here is the result :

Support & Confidence Language

	English	Spanish	French	German	Slavic	Over Indo-E	Chinese	Arabic	J/T/F/M	Other
Support	16	9	8	0	2	10	3	9	2	21
Confidence	0.37	0.42	0.47	0.0	0.5	0.33	0.75	0.47	0.5	0.45

Support & Confidence landmass

	N.America	S.America	Europe	Africa	Asia	Oceania
Support	10	8	4	24	20	14
Confidence	0.3225806	0.4705882	0.1142857	0.4615385	0.5128205	0.7000000

Support & Confidence Religion

	Catholic	Other Christian	Muslim	Buddhist	Hindu	Ethnic	Marxist	Others
Support	13	20	17	3	1	13	10	3
Confidence	0.3250000	0.3333333	0.4722222	0.3750000	0.2500000	0.4814815	0.6666667	0.7500000

Support & Confidence Zone

	NE	SE	SW	NW
Support	37	13	9	21
Confidence	0.4065934	0.4482759	0.5625000	0.3620690

From the previous results we can see that Using Chinese rule give the highest Confidence but not the highest support. We can't depend on that because it's not supported rule (Same goes for best Confidence in religion rules(Others))

From landmass Oceania has 14 for Support & 0.7 for Confidence.

In zone SW zone has the best Confidence but yet not the best support compared to landmass rule.

So here I would go with landmass rule for Oceania. It can be explained as common history in Oceania.

First(f)

For saltires I merged two rules from previous request to get better result here so the rule is English for language & Other Christian for religion. and I got Support :15 , Confidence : 0.41.

Crosses: I picked the rule as (zone:SW,language:English,Crosses;0) and I got Support:4,Confidence:0.66.

Sunstars: I used the previous request tables to help me judge so the rule was (language:English,landmass:Oceania,sunstars;0) the result : Support:9,Confidence : 0.75. I would pick sunstar as a rule (although it's the same result of previous request),But still it's the one with best Support and accuracy I got up to know.

Here is the code used for this task :

```
1 #First (f)
2 testdata <- subset(data, saltires > 0)
3 saltires.EnglishAndOtherChristian.Support <- nrow(subset(data, language == "English" & religion ==
4   "Other Christian" & saltires > 0))
5 saltires.EnglishAndOtherChristian.Confidence <- saltires.EnglishAndOtherChristian.Support /
6   nrow(subset(data, language == "English" & religion == "Other Christian"))
7
8 testdata <- subset(data, crosses > 0)
9 crosses.EnglishAndSW.Support <- nrow(subset(data, zone == "SW" & language == "English" & crosses > 0))
10 crosses.EnglishAndSW.Confidence <- crosses.EnglishAndSW.Support / nrow(subset(data, zone == "SW"
11   & language == "English"))
12 testdata <- subset(data, sunstars > 0)
13 sunsorstars.EnglishAndOceania.Support <- nrow(subset(data, language == "English" & landmass ==
14   "Oceania" & sunstars > 0))
15 sunsorstars.EnglishAndOceania.Confidence <- sunsorstars.EnglishAndOceania.Support / nrow(subset(
16   data, language == "English" & landmass == "Oceania"))
```

First(g)

For this question I'll the feature if there is green color in a flag then it's a Muslim country.

	Predicted positives	Predicted negatives
Labelled positives	26	65
Labelled negatives	10	93

To Change the positive class (as I understood we have to take the opposite feature) The feature become if there is no green color then it's not a Muslim country.

	Predicted positives	Predicted negatives
Labelled positives	93	10
Labelled negatives	65	26

We can notice that the result is the same just the places changed depending on what we consider positive ,negative.

```
1 #First(G)
2 #I USED THE CODE FROM PREVIOUS YEAR PRACTICE SESSION JUST CHANGED THE PARAMETERS
3 # Elements of a confusion matrix
4 true.positives = nrow(subset(data, green == 1 & religion == "Muslim"))
5 true.negatives = nrow(subset(data, green == 0 & religion != "Muslim"))
6 false.positives = nrow(subset(data, green == 0 & religion == "Muslim"))
7 false.negatives = nrow(subset(data, green == 1 & religion != "Muslim"))
8
9 confusion.matrix <- rbind(c(true.positives, false.negatives), c(false.positives, true.
10   negatives))
11 colnames(confusion.matrix) <- c("Predicted positives", "Predicted negatives")
12 rownames(confusion.matrix) <- c("Labelled positives", "Labelled negatives")
13 confusion.matrix
14 #Opposite
15 true.positives = nrow(subset(data, green == 0 & religion != "Muslim"))
16 true.negatives = nrow(subset(data, green == 1 & religion == "Muslim"))
17 false.positives = nrow(subset(data, green == 1 & religion != "Muslim"))
18 false.negatives = nrow(subset(data, green == 0 & religion == "Muslim"))
19
20 confusion.matrix <- rbind(c(true.positives, false.negatives), c(false.positives, true.
21   negatives))
22 colnames(confusion.matrix) <- c("Predicted positives", "Predicted negatives")
23 rownames(confusion.matrix) <- c("Labelled positives", "Labelled negatives")
24 confusion.matrix
```

Second Question

Second(a)

I depended on function sample for this request and here is the code:

```
1 #Second (a)
2 GetSamples<-function(X,n){
3   S <- sample(x = X, size = n, replace = TRUE)
4   print(min(S))
5   print(mean(S))
6   return(S)
7 }
8 values<-c(1:10000)
9 GetSamples(values,10)
10
```

Second(b)

For this task I updated the previous function not to SPAM the screen with output.Used sd() function to calculate the deviation.Here is the code after it the result.

```
1 #Second(b)
2 #write the function without printig not to spam
3 GetSamples2<-function(X,n){
4   S <- sample(x = X, size = n, replace = TRUE)
5   return(S)
6 }
7 #define value containers
8 meanVector<-c(1:1000)
9 minVector<-c(1:1000)
10
11
12 for (i in 1:1000)
13 {
14   #Get sample
15   #n = 1,10,100,1000,10000
16   currentSample = GetSamples2(values,10000)
17   #calculate current min,mean
18   meanVector[i] = mean(currentSample)
19   minVector[i]=min(currentSample)
20 }
21 average.mean=mean(meanVector)
22 average.min=mean(minVector)
23 sd(meanVector)
24 sd(minVector)
25
```

The result of previous code was as following :

Size	Standard Deviation		Average	
	mean	min	mean	min
1	2933.957	2933.957	4973.313	4973.313
10	908.7539	878.4209	5065.1008	952.722
100	289.9641	99.06789	4989.76864	99.765
1000	88.99537	9.717406	5001.901446	10.729
10000	28.593	1.023292	5000.2225452	1.589

Previous table tell us, the larger element we take the more accurate our answer will be.In figure 12 and 13(better view) we can see that the more n size increase the less deviation we get.

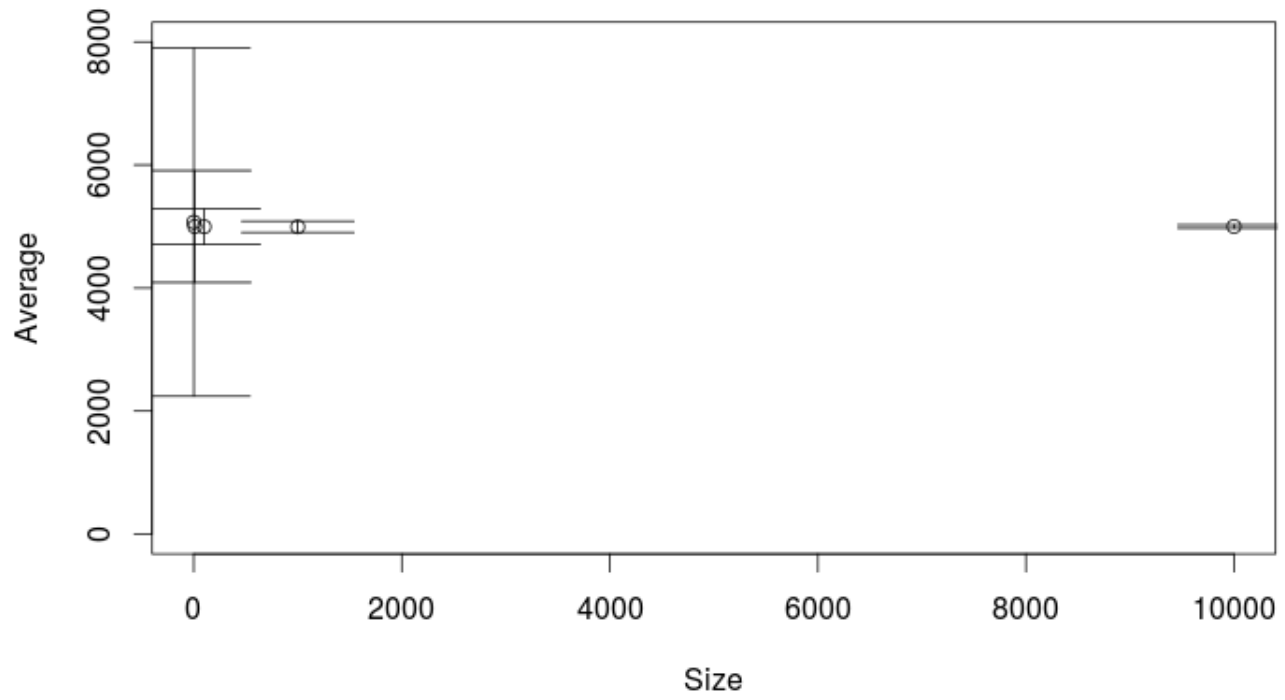


Figure 12: Shows the deviation of mean average vs n size

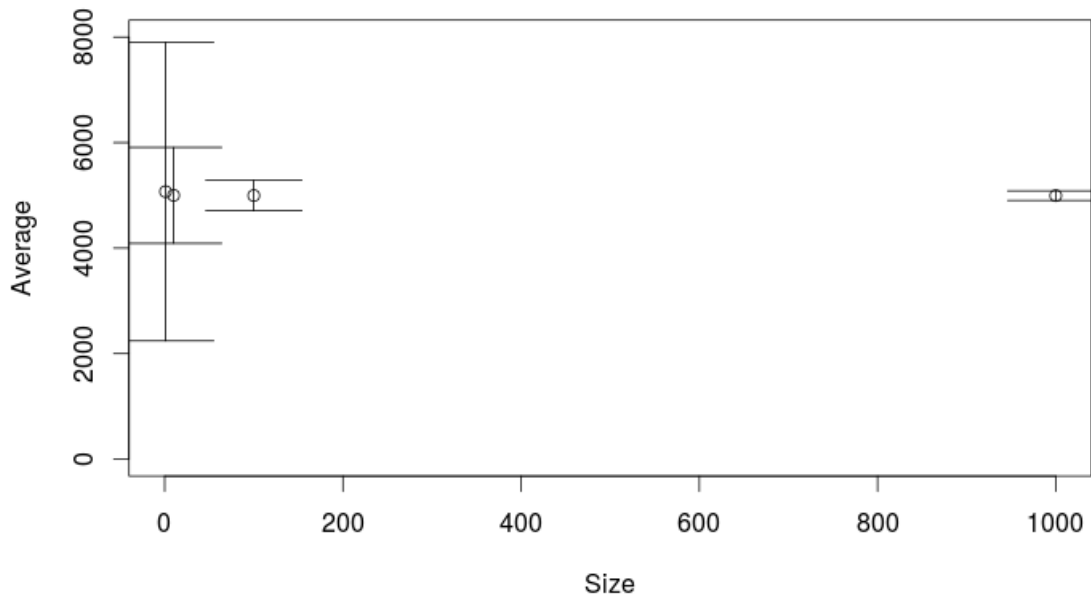


Figure 13: More clear view of deviation over mean average vs n size

The same for min value as shown in figure 14,15(better view), the more samples we take the less deviation we have.

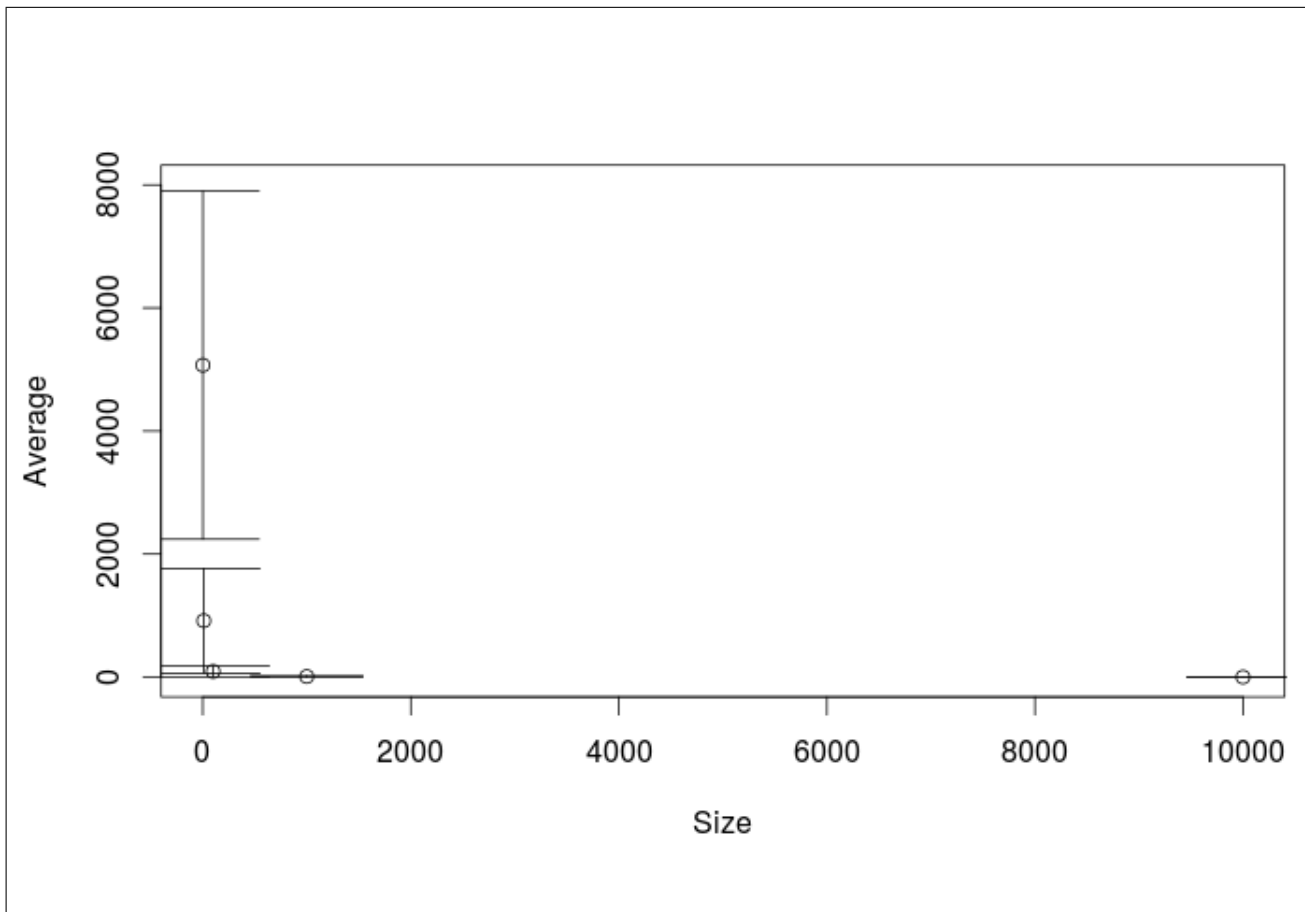


Figure 14: Shows deviation over min average vs n size

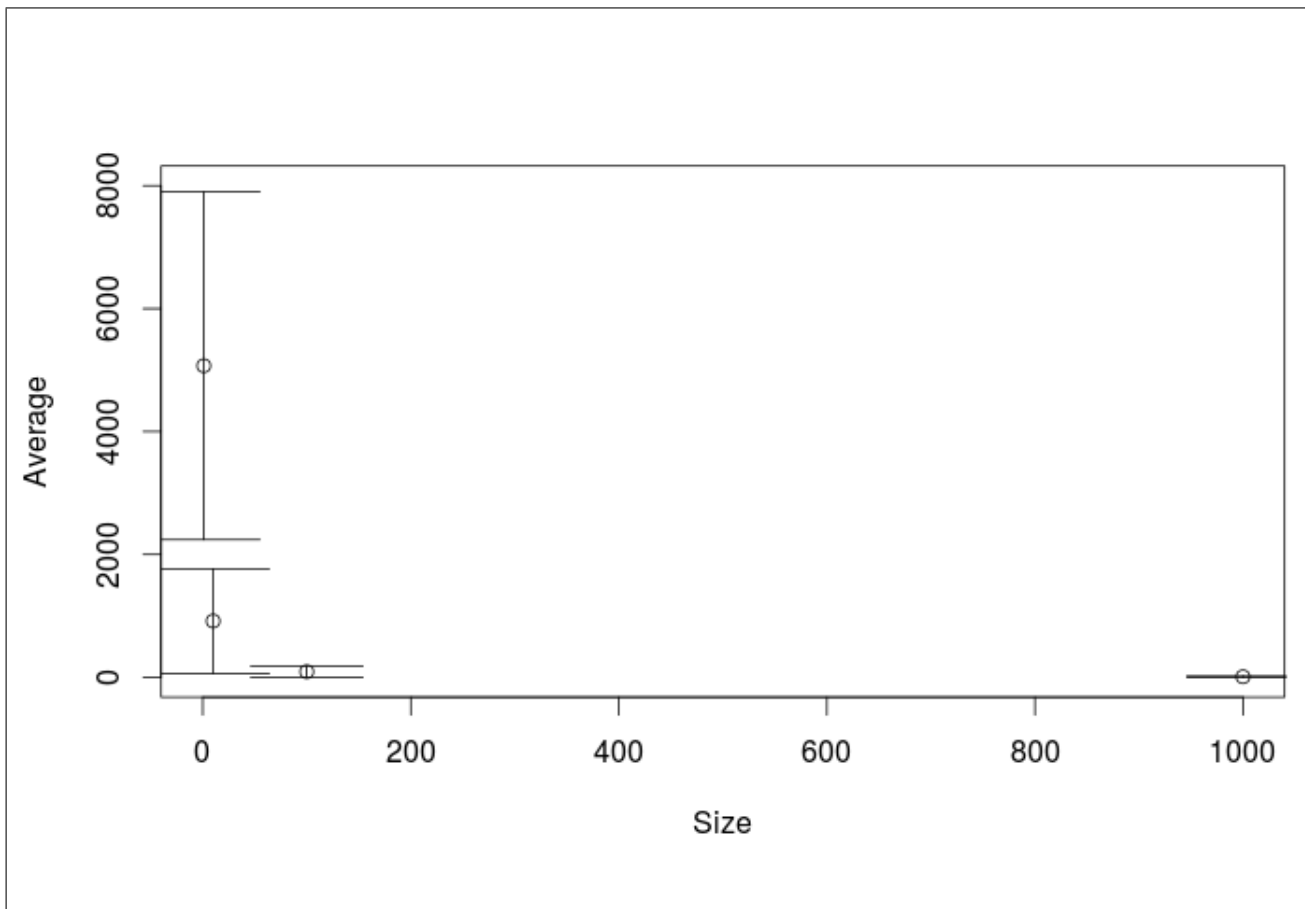


Figure 15: Same as figure 14 but focusing on first 4 n sizes

Here is the Code is used for this task:

```

1 #Second(b)
2 values<-c(1:10000)
3 #write the function without printig not to spam
4 GetSamples2<-function(X,n){
5   S <- sample(x = X,size = n,replace = TRUE)
6   return (S)
7 }
8 #define value containers
9 meanVector<-c(1:1000)
10 minVector<-c(1:1000)
11 meav <- vector(mode="numeric", length=0)
12 miav <- vector(mode="numeric", length=0)
13 mesd <- vector(mode="numeric", length=0)
14 misd <- vector(mode="numeric", length=0)
15 for (n in c(1,10,100,1000,10000))
16 {
17
18   for (i in 1:1000)
19   {
20     #Get sample
21     #n = 1,10,100,1000,10000
22     currentSample = GetSamples2(values,n)
23     #calculate current min,mean
24     meanVector[i] = mean(currentSample)
25     minVector[i]=min(currentSample)
26   }
27   meav <- c(meav,mean(meanVector))
28   miav <- c(miav,mean(minVector))
29   mesd <- c(mesd,sd(meanVector))
30   misd <- c(misd,sd(minVector))
31
32 }
33 #THIS CODE WHERE COPIED FROM THIS SOURCE :http://stackoverflow.com/questions/15063287/add-
34   error-bars-to-show-standard-deviation-on-a-plot-in-r
35 #I CHANGED THE VALUES TO FIT WHATS REQUESTED IN THE QUESTION
36 #prepare data for (mean , min one of them each time )
37 d = data.frame(
38   x = c(1,10,100,1000,10000)
39   , y = miav
40   , sd = misd
41 )
42 ##install.packages("Hmisc", dependencies=T)
43 library("Hmisc")
44
45 # add error bars (without adjusting yrange)
46 plot(d$x, d$y, type="n",ylim=c(0,8000),xlim = c(0,1000),xlab = "Size",ylab = "Average")
47 with (
48   data = d
49   , expr = errbar(x, y, y+sd, y-sd, add=T, pch=1, cap=.1)
50 )
51

```

Second(c)

For this task I used the exact code from previous task after changing the sample function as following :

```

1 GetSamples2<-function(X,n){
2   S <- sample(x = X,size = n,replace = FALSE)
3   return (S)
4 }
5

```

Firstly here is the table of the results :

	Standard Deviation		Average	
Size	mean	min	mean	min
1	2909.84151	2909.841512	4959.063	4959.063
10	878.72027	793.918030	4983.674	889.157
100	293.56768	91.638599	5006.395	97.378
1000	86.95006	9.627084	4997.647	10.044
10000	0.00000	0.000000	5000.500	1.000

Using sampling with out replacement get better results and less deviation in most cases.The following is the visualization with error bars.Two figures for each (mean,min) one with all n sizes, and the other one after minimizing x scope to get better view.

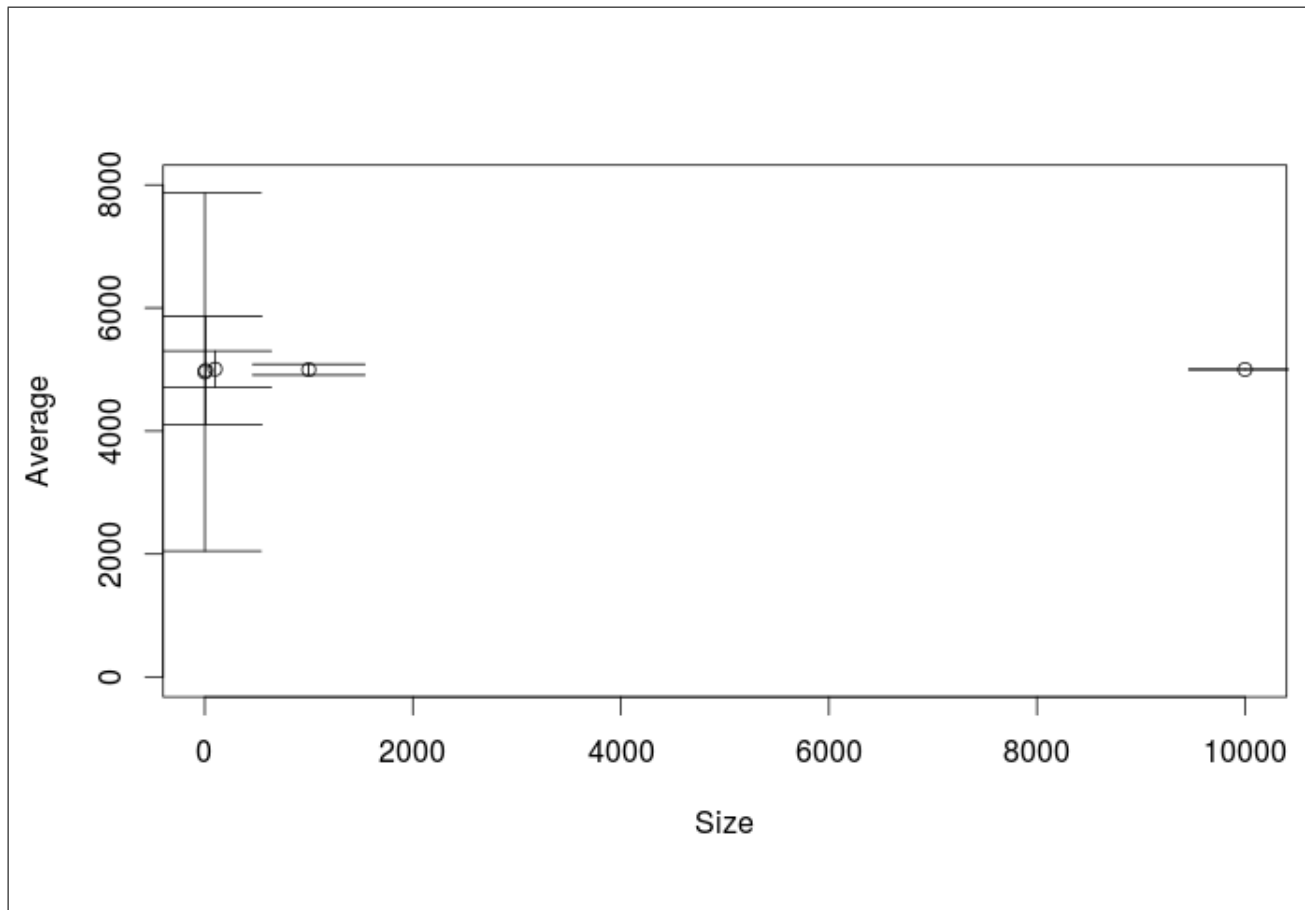


Figure 16: Deviation (mean average vs n size)without replacement

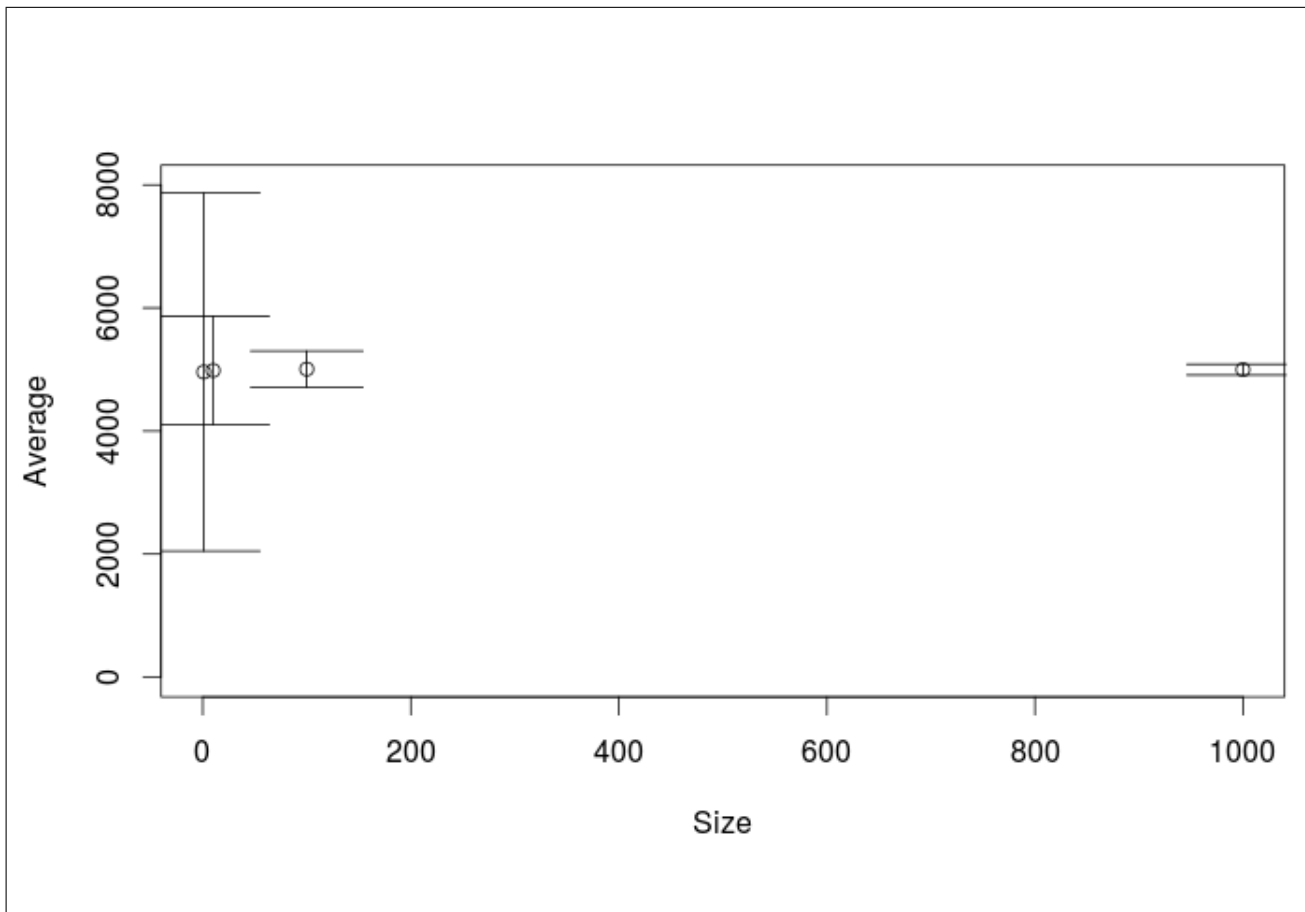


Figure 17: Deviation (mean average vs n size)without replacement(clearer view)

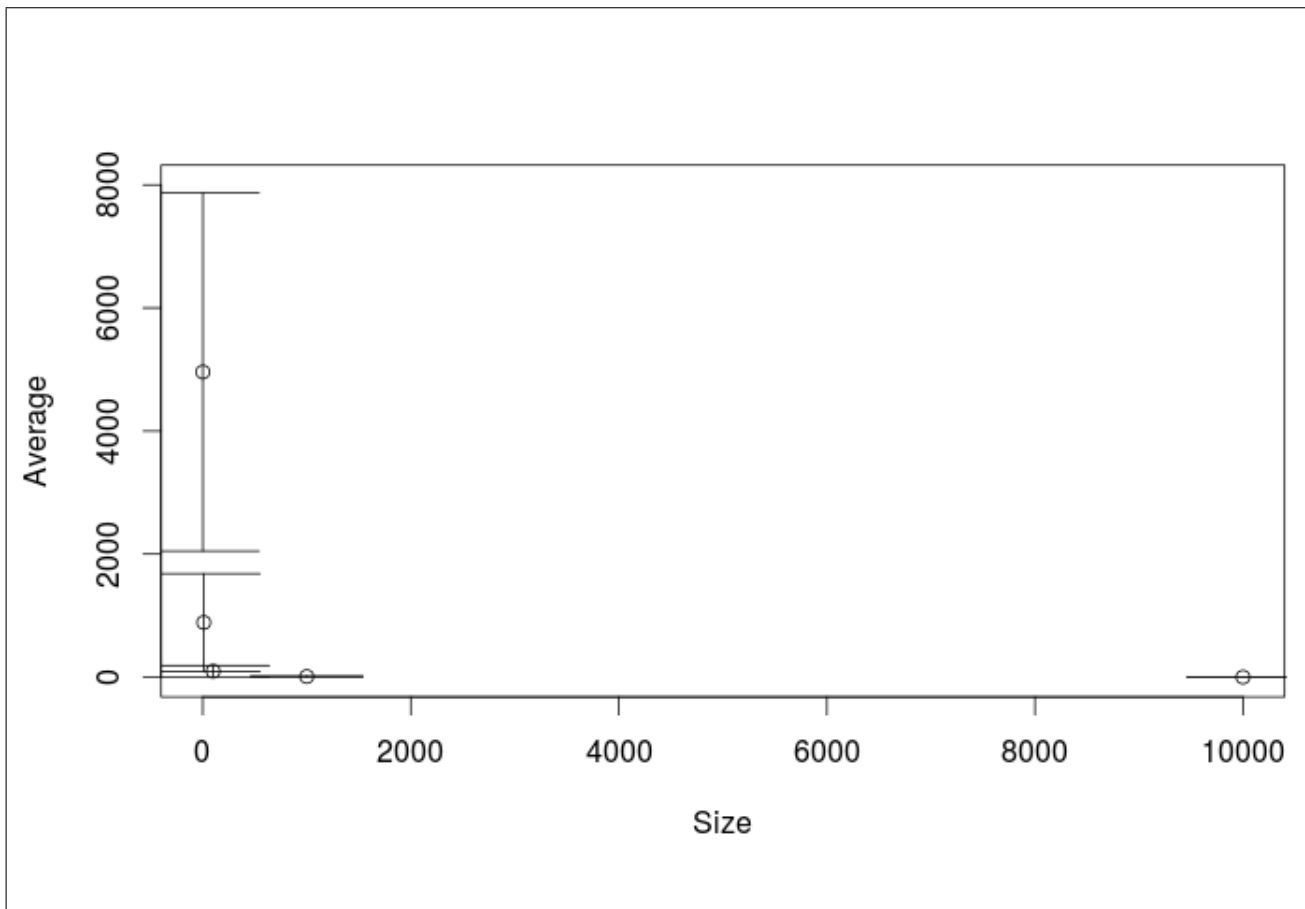


Figure 18: Deviation (min average vs n size)without replacement

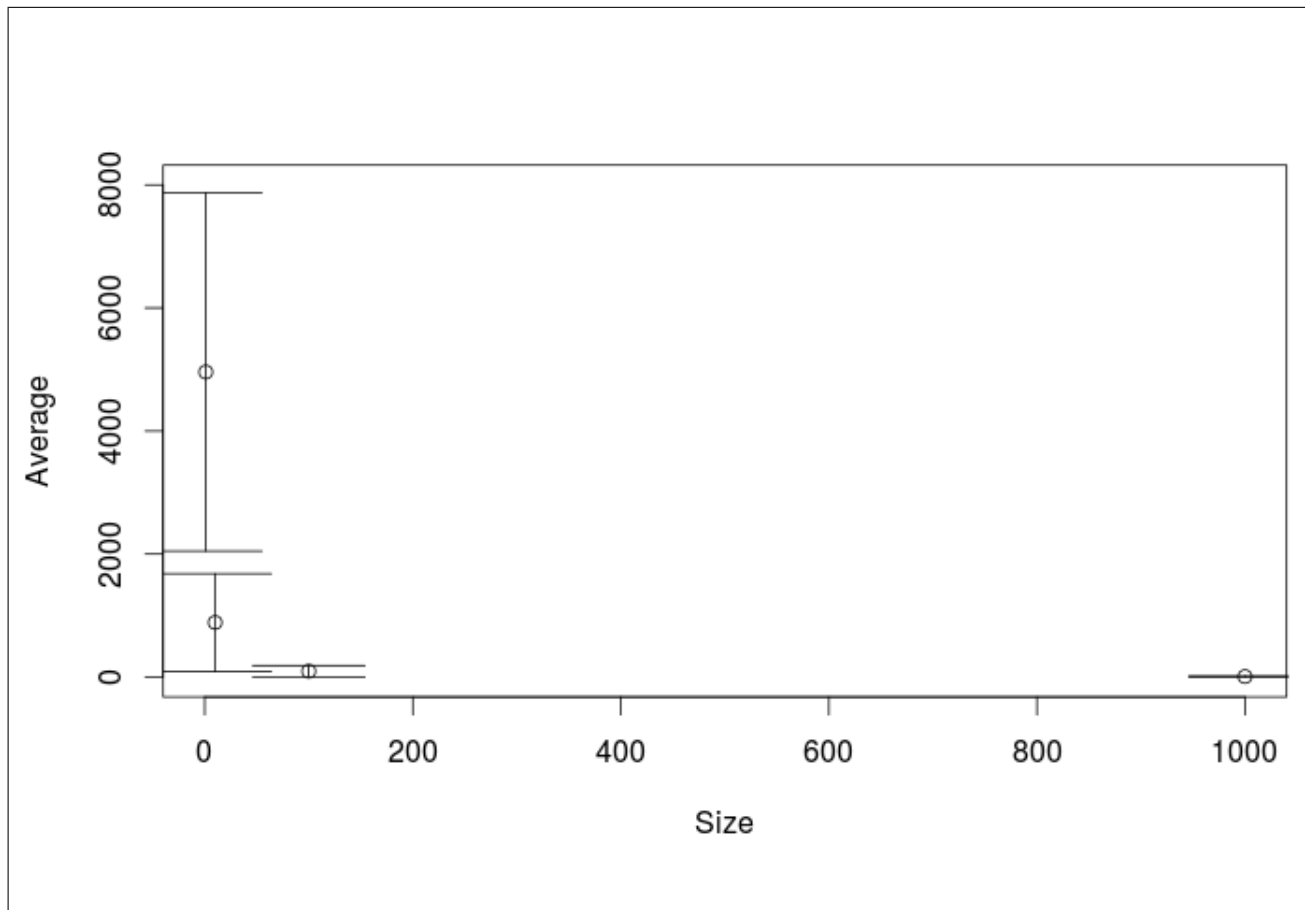


Figure 19: Deviation (min average vs n size)without replacement(clearer view)

Comparing the previous figures with ones in (Second question (b)) won't show much difference. That's because the difference is small compared to the scale of the axis. But the comparison by table values is clear.

Third Question

Third(a)

The Code for this question :

```

1 #Third Question (a)
2 rm (list = ls())
3 RateFunction<-function (years=20)
4 {
5   rates<-vector(mode = "numeric",length = 0)
6   mean=2
7   sd =0.5
8   for (i in c(1:years))
9   {
10    #this is not the first round
11    if (i>1)
12    {
13     #If previous rate larger or equal 1 mean =2
14     if (rates[i-1]>=1)
15     {
16      mean=2
17     }
18     #If Previous rate less than 1
19     else if (rates[i-1]<1)
20     {
21      mean =0
22     }
23    }
24    #calculate the new rate.
25    print (mean)
26    currentrate = rnorm(1,mean = mean,sd = sd)

```

```

27 #Fix the current Rate :
28 if (currentrate < 0)
29 {
30   currentrate = 0
31 }
32 #Fix the rate
33 rates <- c(rates, currentrate)
34 }
35 return (rates)
36 }
37 plot(c(1:20), RateFunction(years = 20), type = "l")
38

```

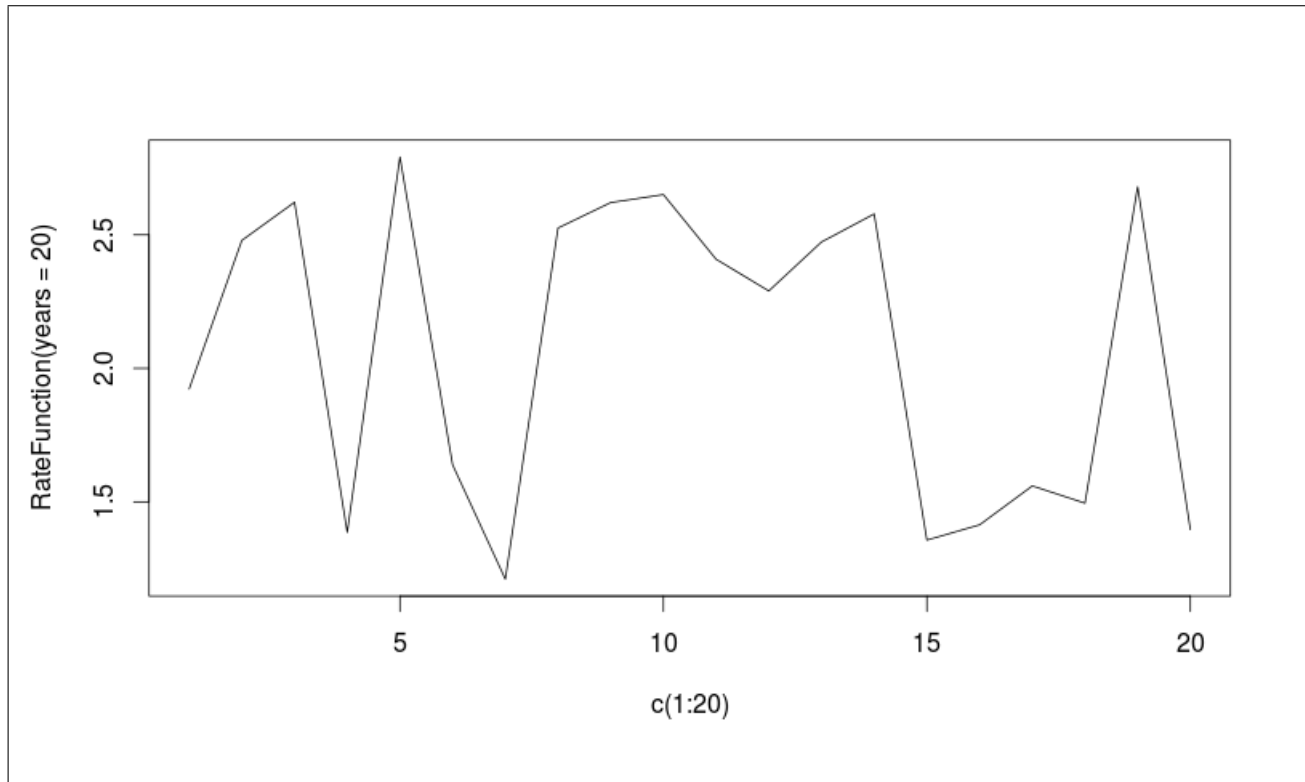


Figure 20: The Rate value for each year

Third(b)

Under Construction :D

Third(c)

Third(d)

Fourth Question

Fourth(a)

Fourth(b)

Fourth(c)