Data Mining- Lab Exam

Time: 24 hours
Marks:100

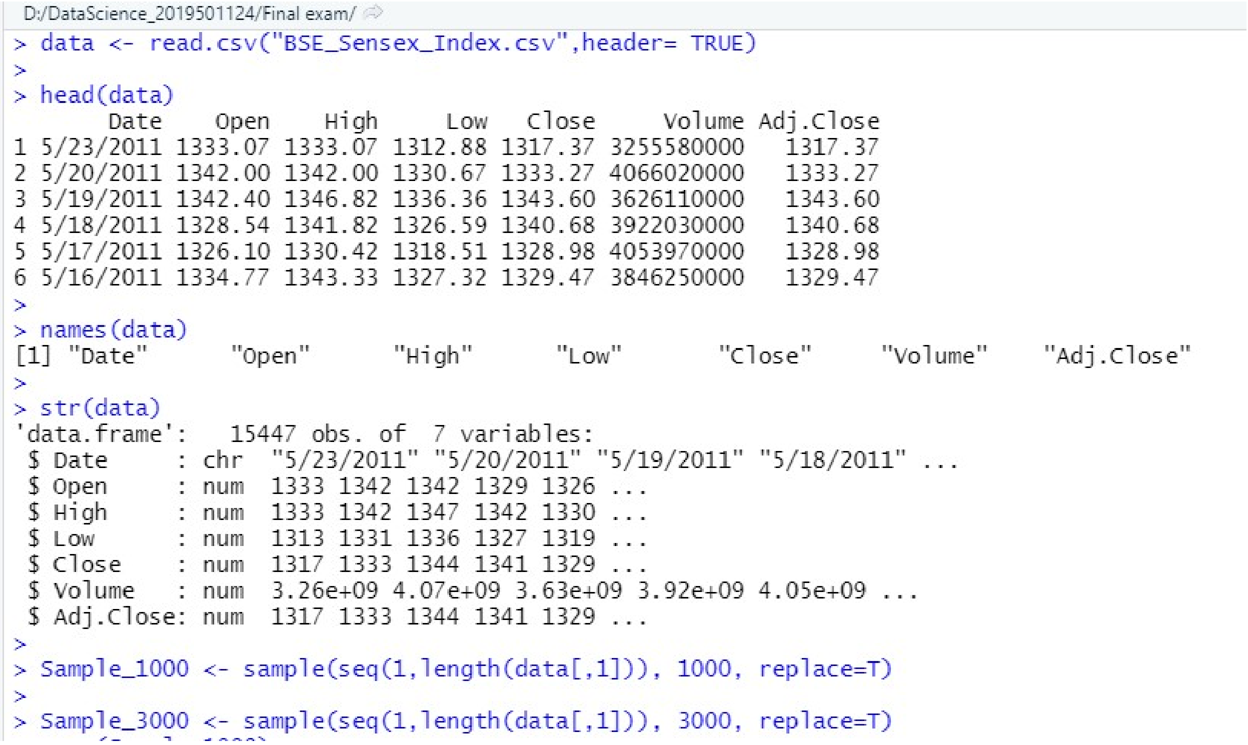
Open a document and update document with your answers for each question and submit it.

1. a) For the dataset BSE_Sensex_Index.csv, create an extra column of successive differences for each column of numeric values in this data file. Extract two simple random samples with replacement of 1000 and 3000 observations (rows). Show your R commands for doing this.
   Do the same thing by using Excel. Show your Excel commands.
   Note: Successive difference for date d1= (date d1 value-immediate available previous date of d1 value)/immediate available previous date of d1. For the last row fill up values with mean of its immediate three previous row values.

   data<-read.csv ("BSE_Sensex_Index.csv", header=TRUE)

   sample_1000 <- sample (seq (1, length (data [,1])), 1000, replace = T)

   sample_3000 <- sample (seq (1, length (data [,1])), 3000, replace = T)

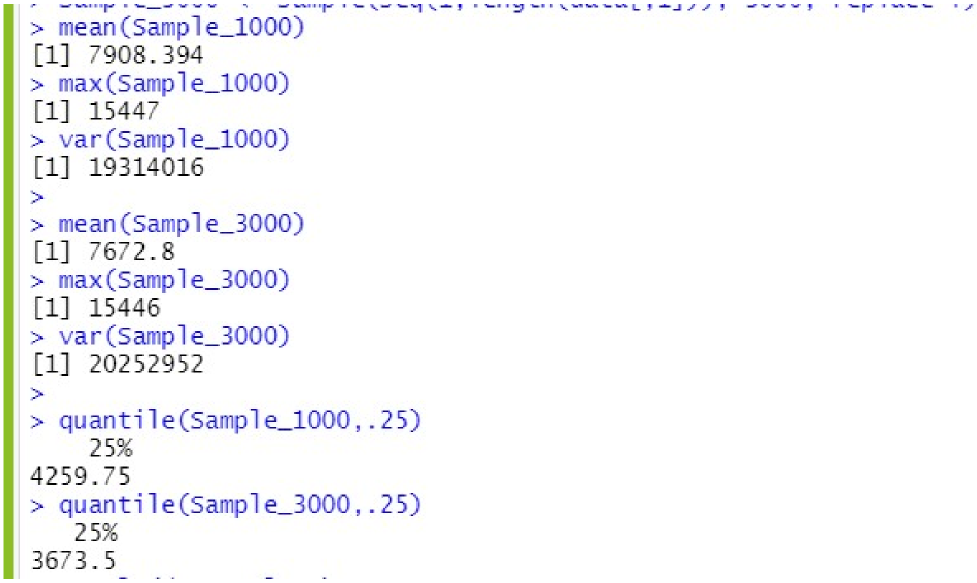


```
D:/DataScience_2019501124/Final exam/
> data <- read.csv("BSE_Sensex_Index.csv",header= TRUE)
>
> head(data)
       Date    Open    High     Low   Close     Volume Adj.Close
1 5/23/2011 1333.07 1333.07 1312.88 1317.37 3255580000   1317.37
2 5/20/2011 1342.00 1342.00 1330.67 1333.27 4066020000   1333.27
3 5/19/2011 1342.40 1346.82 1336.36 1343.60 3626110000   1343.60
4 5/18/2011 1328.54 1341.82 1326.59 1340.68 3922030000   1340.68
5 5/17/2011 1326.10 1330.42 1318.51 1328.98 4053970000   1328.98
6 5/16/2011 1334.77 1343.33 1327.32 1329.47 3846250000   1329.47
>
> names(data)
[1] "Date"      "Open"      "High"      "Low"       "Close"     "Volume"    "Adj.Close"
>
> str(data)
'data.frame':   15447 obs. of  7 variables:
 $ Date     : chr  "5/23/2011" "5/20/2011" "5/19/2011" "5/18/2011" ...
 $ Open     : num  1333 1342 1342 1329 1326 ...
 $ High     : num  1333 1342 1347 1342 1330 ...
 $ Low      : num  1313 1331 1336 1327 1319 ...
 $ Close    : num  1317 1333 1344 1341 1329 ...
 $ Volume   : num  3.26e+09 4.07e+09 3.63e+09 3.92e+09 4.05e+09 ...
 $ Adj.Close: num  1317 1333 1344 1341 1329 ...
>
> Sample_1000 <- sample(seq(1,length(data[,1])), 1000, replace=T)
>
> Sample_3000 <- sample(seq(1,length(data[,1])), 3000, replace=T)
```

b) For your samples, use the functions mean(), max(), var() and quartile(,.25) to compute the mean, maximum, variance and 1st quartile respectively for each column which has successive differences. Show your R code and the resulting values.
Do the same thing by using Excel. Show your Excel commands.

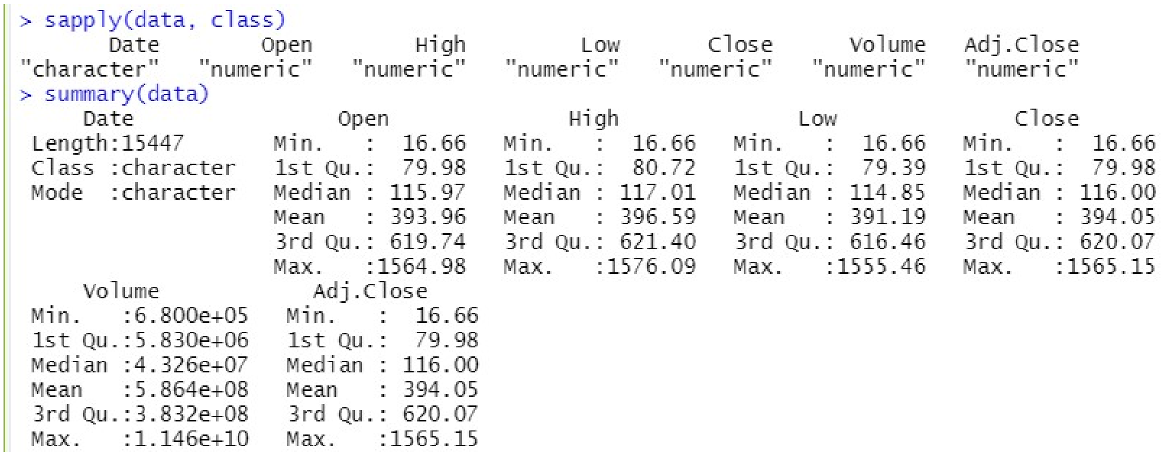
mean(sample_1000)	**7908.394**
max(sample_1000)	**15447**
var(sample_1000)	**19314016**
quantile(sample_1000,.25)	**4259.75**

mean(sample_3000)	**7672.8**
max(sample_3000)	**15446**
var(sample_3000)	**2025952**
quantile(sample_3000,.25)	**3673.5**



```
> mean(Sample_1000)
[1] 7908.394
> max(Sample_1000)
[1] 15447
> var(Sample_1000)
[1] 19314016
>
> mean(Sample_3000)
[1] 7672.8
> max(Sample_3000)
[1] 15446
> var(Sample_3000)
[1] 20252952
>
> quantile(Sample_1000,.25)
    25%
4259.75
> quantile(Sample_3000,.25)
   25%
3673.5
```

c) Compute the same quantities in part b on the entire data set and show your answers. How much do they differ from your answers in part b? Do you find any significant difference between two sample values like mean in comparison with entire data? If so what explanation you can give for that?
Do the same thing by using Excel. Show your Excel commands.

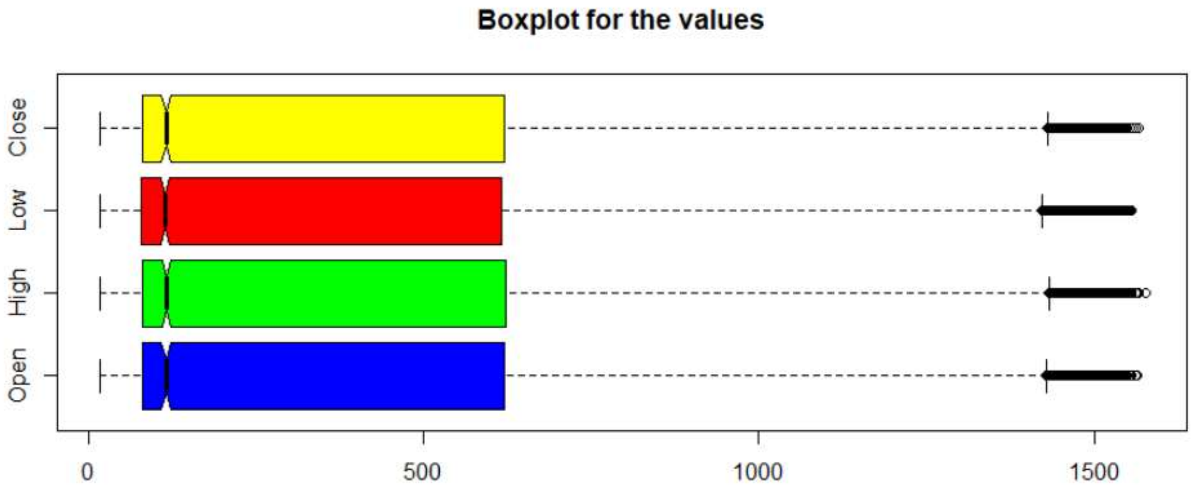


```
> sapply(data, class)
       Date        Open        High         Low       Close      Volume   Adj.Close
"character"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
> summary(data)
     Date                Open              High              Low              Close
 Length:15447       Min.   :  16.66   Min.   :  16.66   Min.   :  16.66   Min.   :  16.66
 Class :character   1st Qu.:  79.98   1st Qu.:  80.72   1st Qu.:  79.39   1st Qu.:  79.98
 Mode  :character   Median : 115.97   Median : 117.01   Median : 114.85   Median : 116.00
                    Mean   : 393.96   Mean   : 396.59   Mean   : 391.19   Mean   : 394.05
                    3rd Qu.: 619.74   3rd Qu.: 621.40   3rd Qu.: 616.46   3rd Qu.: 620.07
                    Max.   :1564.98   Max.   :1576.09   Max.   :1555.46   Max.   :1565.15
     Volume            Adj.Close
 Min.   :6.800e+05   Min.   :  16.66
 1st Qu.:5.830e+06   1st Qu.:  79.98
 Median :4.326e+07   Median : 116.00
 Mean   :5.864e+08   Mean   : 394.05
 3rd Qu.:3.832e+08   3rd Qu.: 620.07
 Max.   :1.146e+10   Max.   :1565.15
```

d) Use R to produce a single graph displaying a boxplot for open, close, high and low. Include the R commands and the plot.

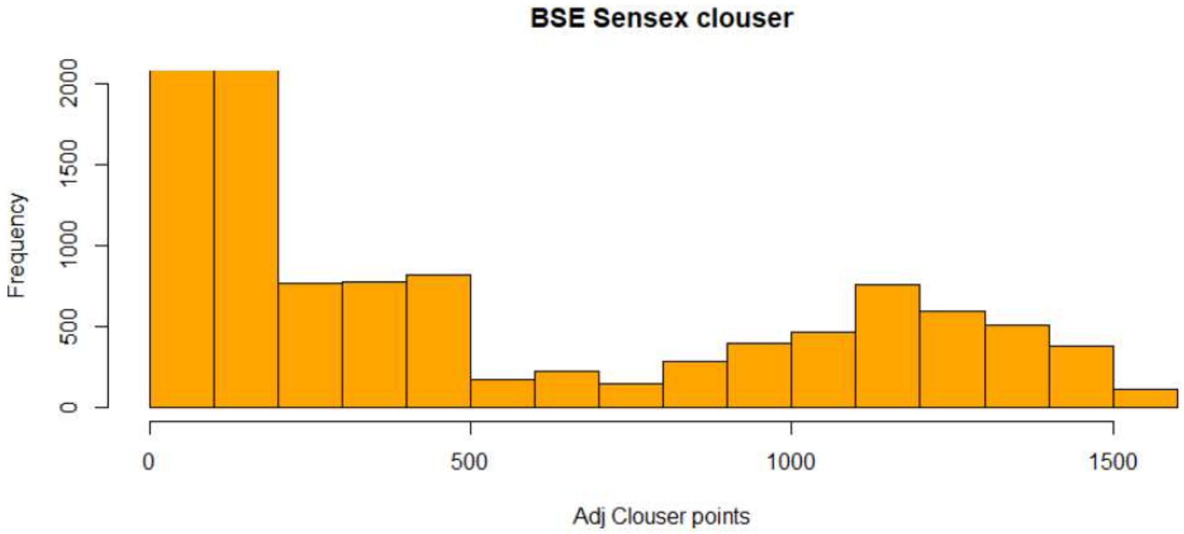
Do the same thing by using Excel. Show your Excel commands

```
open <- data$Open
high <- data$High
low <- data$Low
close <- data$Close

boxplot (open, high, low, close,
    main = "Boxplot for the values",
    names = c ("Open", "High", "Low", "Close"),
    col = c ("blue", "green","red","yellow"),
    horizontal = TRUE,
    notch = TRUE)
```



e) Use R to produce a frequency histogram for Close values. Use intervals of width 2000 beginning at 0. Include the R commands and the plot.
Do the same thing by using Excel. Show your Excel commands. (10+10=20M)

hist(data$Close, ylim = c(0,2000), col="orange”, main = "BSE Sensex clouser", xlab="Adj Clouser points")

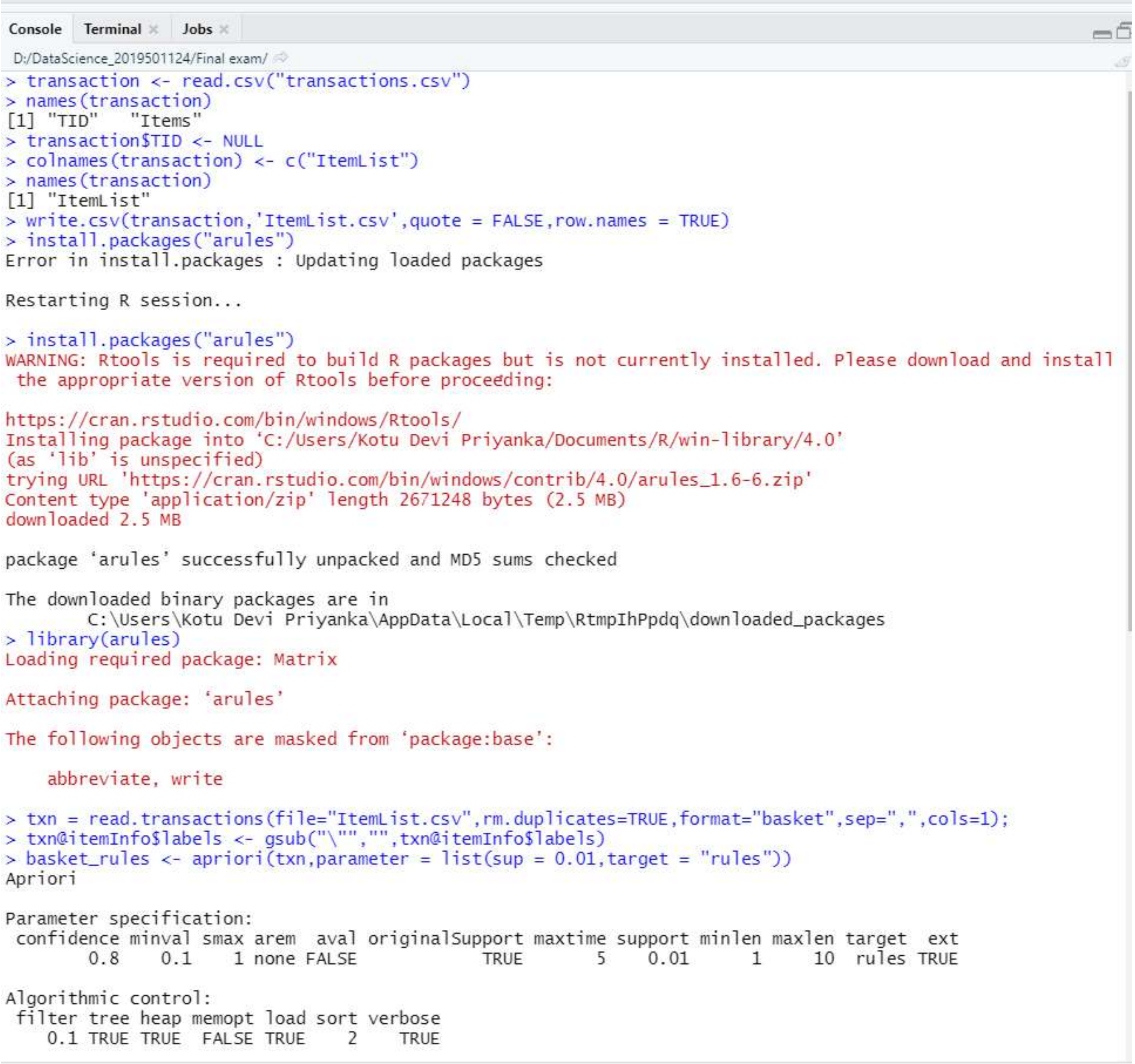


2. Implement Apriori Algorithm or use built in packages to find out the frequent itemsets and generate rules for frequent itemsets. Trace and submit the program output for the following given dataset of transactions with a minimum support of 3. (10M)

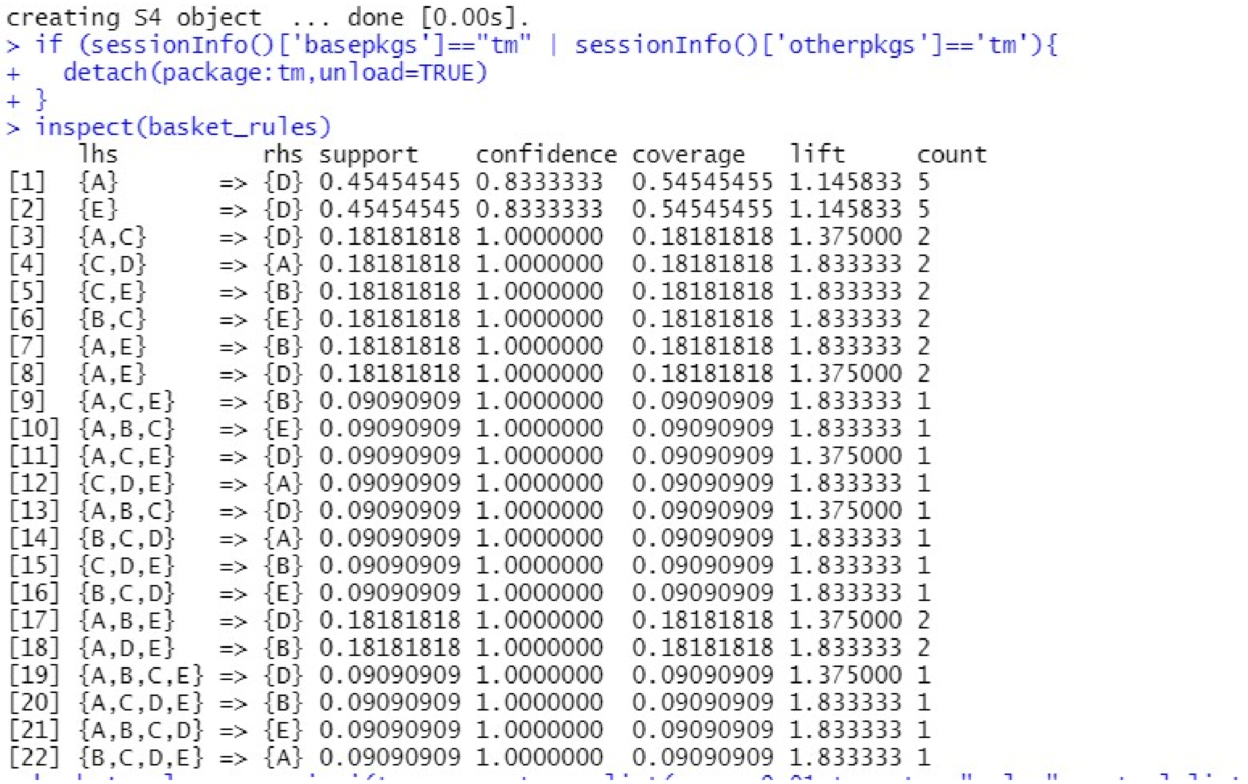
```
TID, Items
101, A,B,C,D,E
102, A,C,D
103, D,E
104, B,C,E
105, A,B,D,E
106, A,B
107, B,D,E
108, A,B,D
109, A,D
110, D,E
```

```
#Created a csv file for the given data
transaction <- read.csv("transactions.csv")

#In order to know the column names
names(transaction)
transaction$TID <- NULL
colnames(transaction) <- c("ItemList")

write.csv (transaction,'ItemList.csv', quote = FALSE, row.names = TRUE)
```

```
#Itemslist.csv
#rm. duplicates – to confirm whether there are duplicates or not
#format-basket (row 1: TID, row 2: List of items)
# sep - separator (“,”)
#cols – column number of TID

txn =
read.transactions(file="ItemList.csv",rm.duplicates=TRUE,format="basket",sep=",",cols=1);

#installing the required packages to perform operations using apriori algorithm

install.packages("arules")
library(arules)

txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)
basket_rules <- apriori(txn,parameter = list(sup = 0.01,target = "rules"))

if (sessionInfo() ['basepkgs']=="tm" | sessionInfo()['otherpkgs']=='tm'){
  detach(package:tm,unload=TRUE)
}
inspect(basket_rules)
```



```
> transaction <- read.csv("transactions.csv")
> names(transaction)
[1] "TID"   "Items"
> transaction$TID <- NULL
> colnames(transaction) <- c("ItemList")
> names(transaction)
[1] "ItemList"
> write.csv(transaction,'ItemList.csv',quote = FALSE,row.names = TRUE)
> install.packages("arules")
Error in install.packages : Updating loaded packages

Restarting R session...

> install.packages("arules")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install
 the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/Kotu Devi Priyanka/Documents/R/win-library/4.0’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/arules_1.6-6.zip'
Content type 'application/zip' length 2671248 bytes (2.5 MB)
downloaded 2.5 MB

package ‘arules’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
	C:\Users\Kotu Devi Priyanka\AppData\Local\Temp\RtmpIhPpdq\downloaded_packages
> library(arules)
Loading required package: Matrix

Attaching package: ‘arules’

The following objects are masked from ‘package:base’:

    abbreviate, write

> txn = read.transactions(file="ItemList.csv",rm.duplicates=TRUE,format="basket",sep=",",cols=1);
> txn@itemInfo$labels <- gsub("\"","",txn@itemInfo$labels)
> basket_rules <- apriori(txn,parameter = list(sup = 0.01,target = "rules"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
```



```
creating S4 object  ... done [0.00s].
> if (sessionInfo()['basepkgs']=="tm" | sessionInfo()['otherpkgs']=='tm'){
+   detach(package:tm,unload=TRUE)
+ }
> inspect(basket_rules)
     lhs          rhs support    confidence coverage   lift     count
[1]  {A}       => {D} 0.45454545 0.8333333  0.54545455 1.145833 5
[2]  {E}       => {D} 0.45454545 0.8333333  0.54545455 1.145833 5
[3]  {A,C}     => {D} 0.18181818 1.0000000  0.18181818 1.375000 2
[4]  {C,D}     => {A} 0.18181818 1.0000000  0.18181818 1.833333 2
[5]  {C,E}     => {B} 0.18181818 1.0000000  0.18181818 1.833333 2
[6]  {B,C}     => {E} 0.18181818 1.0000000  0.18181818 1.833333 2
[7]  {A,E}     => {B} 0.18181818 1.0000000  0.18181818 1.833333 2
[8]  {A,E}     => {D} 0.18181818 1.0000000  0.18181818 1.375000 2
[9]  {A,C,E}   => {B} 0.09090909 1.0000000  0.09090909 1.833333 1
[10] {A,B,C}   => {E} 0.09090909 1.0000000  0.09090909 1.833333 1
[11] {A,C,E}   => {D} 0.09090909 1.0000000  0.09090909 1.375000 1
[12] {C,D,E}   => {A} 0.09090909 1.0000000  0.09090909 1.833333 1
[13] {A,B,C}   => {D} 0.09090909 1.0000000  0.09090909 1.375000 1
[14] {B,C,D}   => {A} 0.09090909 1.0000000  0.09090909 1.833333 1
[15] {C,D,E}   => {B} 0.09090909 1.0000000  0.09090909 1.833333 1
[16] {B,C,D}   => {E} 0.09090909 1.0000000  0.09090909 1.833333 1
[17] {A,B,E}   => {D} 0.18181818 1.0000000  0.18181818 1.375000 2
[18] {A,D,E}   => {B} 0.18181818 1.0000000  0.18181818 1.833333 2
[19] {A,B,C,E} => {D} 0.09090909 1.0000000  0.09090909 1.375000 1
[20] {A,C,D,E} => {B} 0.09090909 1.0000000  0.09090909 1.833333 1
[21] {A,B,C,D} => {E} 0.09090909 1.0000000  0.09090909 1.833333 1
[22] {B,C,D,E} => {A} 0.09090909 1.0000000  0.09090909 1.833333 1
```

Step 1

K=1

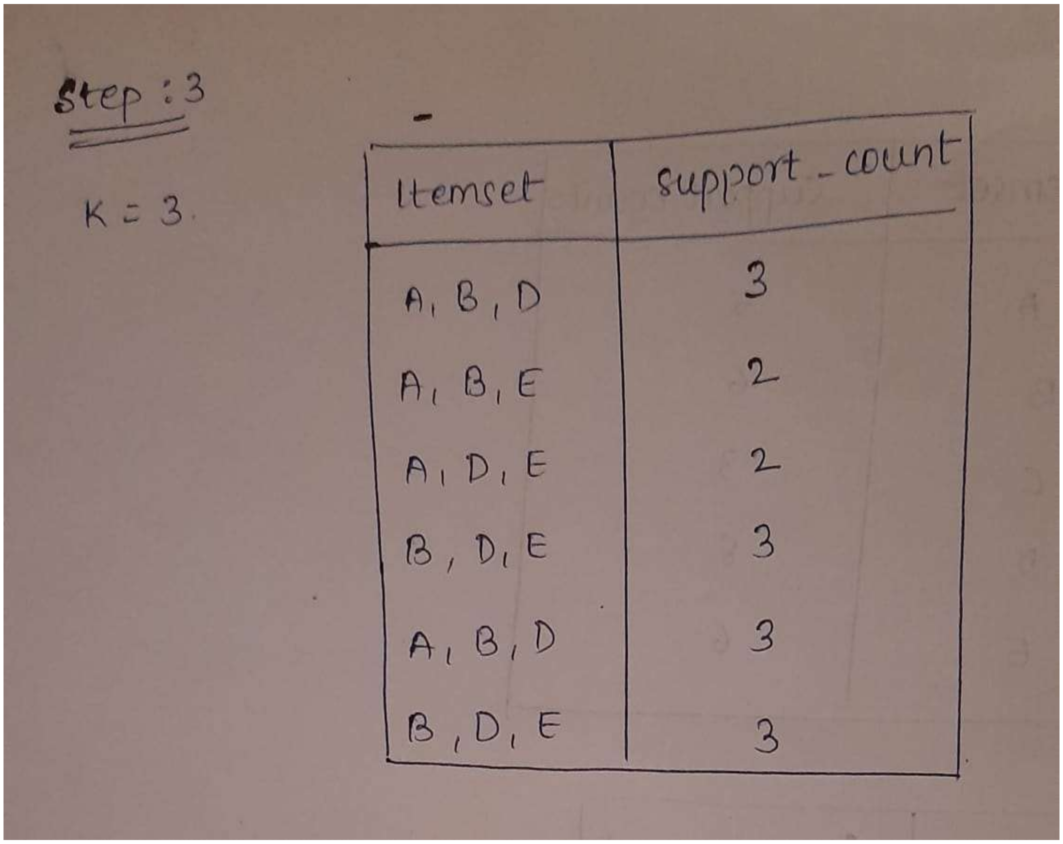
Itemset	Support count
A	5
B	6
C	3
D	8
E	6

K=2

Itemset	Support_count
A, B	4
A, C	2
A, D	5
A, E	2
B, C	2
B, D	4
B, E	4
C, D	2
C, E	2
D, E	5

Minimum suppott count : 3

Itemset	support_count
A, B	4
A, D	5
B, D	4
B, E	4
D, E	5



Step : 3

K = 3.

Itemset	support_count
A, B, D	3
A, B, E	2
A, D, E	2
B, D, E	3
A, B, D	3
B, D, E	3

Association Rules:

{A, B} ➔ {D} ; confidence = 4/8 = 50%
{A, D} ➔ {B} ; confidence = 5/6 = 83%
{B, D} ➔ {A} ; confidence = 4/5 = 80%
{A} ➔ {B, D} ; confidence = 3/5 = 60%
{B} ➔ {A, D} ; confidence = 3/6 = 50%
{D} ➔ {A, B} ; confidence = 3/8 = 37.5%

If minimum confidence is 50% then first 5 rules can be considered.
We can find rules for {B, D, E} and other item sets similarly.

3. Build Decision Trees by using i) information gain and ii) misclassification error rate for Lenses Data Set provided at http://archive.ics.uci.edu/ml/datasets/Lenses. In terms of tree size what do you conclude comparing these two? (10M)

4. Fit 1, 2 and 3-nearest-neighbor classifiers to the Liver Disorders Data Set at http://archive.ics.uci.edu/ml/datasets/Liver+Disorders for measures Euclidean and cosine.

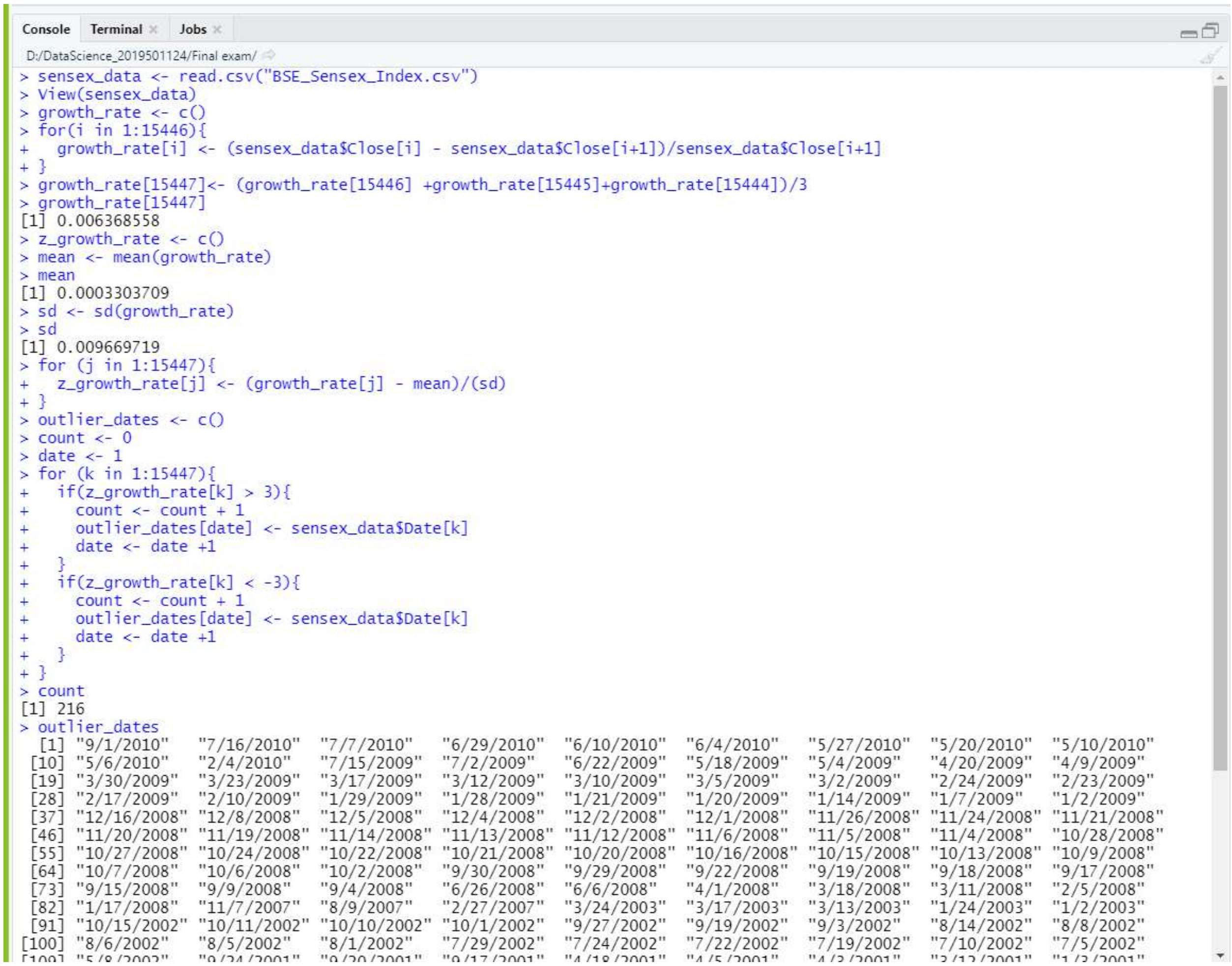
Last but one column is a decision attribute. Replace decision values in to 4 classes (0<=c1<5, 5<=c2<10, 10<=c3<15, 15<=c4<=20). Last column is a data split column in to training and test sets. 1 means the object is used for training. 2 means the object is used for testing. Explain the input parameters you provided for the classifier. Compute the misclassification error on the training data and also on the test data. Annotate your program. (10M)

5. Use Support Vector machine for above problem. And compare the performance of both. Explain the input parameters you provided for the classifier. (10M)

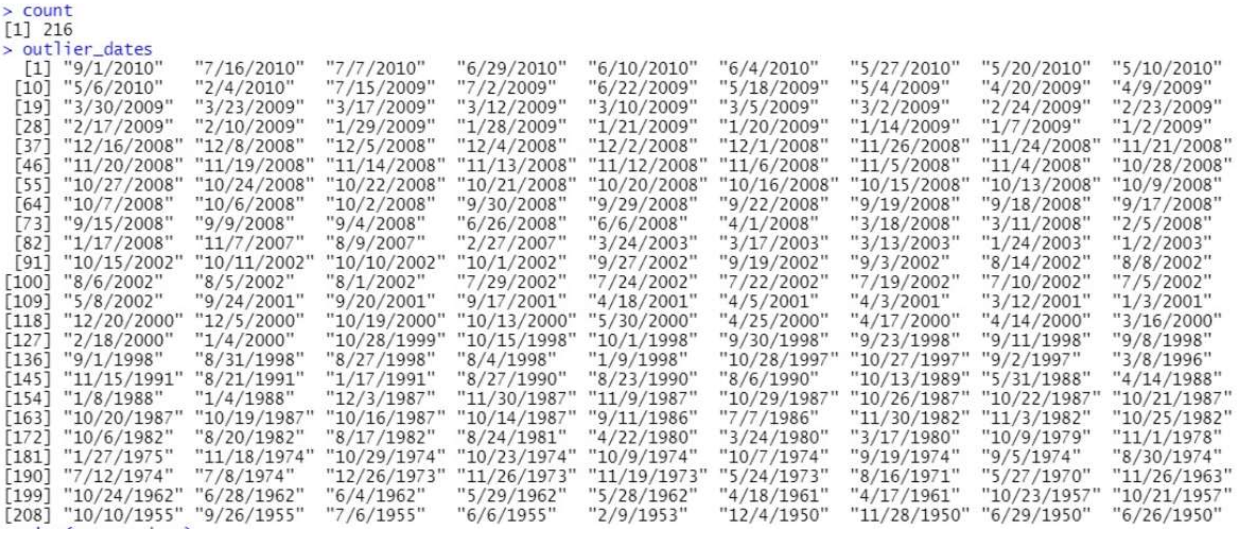
6. Create k-means clusters for k=4 for the Liver Disorders Data Set at http://archive.ics.uci.edu/ml/datasets/Liver+Disorders . Explain the input parameters you provided for the clustering algorithm. Plot the fitted cluster centers using a different color. Finally assign the cluster membership for the points to the nearest cluster center. Color the points according to their cluster membership. (10+10=20M)

7. Compute the misclassification error that would result if you used your clustering rule to classify the data by assigning the majority class of the cluster. (10M)

8. Consider the dataset BSE_Sensex_Index.csv. Create an extra column of successive growth rate for column close where the successive growth rate is defined as
(value of day x- value of day x-1)/value of day x-1. Use a z score cut off of 3 to identify any outliers. List the respective dates from the csv file on which day these outliers fall. (10M)



```
> sensex_data <- read.csv("BSE_Sensex_Index.csv")
> View(sensex_data)
> growth_rate <- c()
> for(i in 1:15446){
+   growth_rate[i] <- (sensex_data$Close[i] - sensex_data$Close[i+1])/sensex_data$Close[i+1]
+ }
> growth_rate[15447]<- (growth_rate[15446] +growth_rate[15445]+growth_rate[15444])/3
> growth_rate[15447]
[1] 0.006368558
> z_growth_rate <- c()
> mean <- mean(growth_rate)
> mean
[1] 0.0003303709
> sd <- sd(growth_rate)
> sd
[1] 0.009669719
> for (j in 1:15447){
+   z_growth_rate[j] <- (growth_rate[j] - mean)/(sd)
+ }
> outlier_dates <- c()
> count <- 0
> date <- 1
> for (k in 1:15447){
+   if(z_growth_rate[k] > 3){
+     count <- count + 1
+     outlier_dates[date] <- sensex_data$Date[k]
+     date <- date +1
+   }
+   if(z_growth_rate[k] < -3){
+     count <- count + 1
+     outlier_dates[date] <- sensex_data$Date[k]
+     date <- date +1
+   }
+ }
> count
[1] 216
> outlier_dates
  [1] "9/1/2010"   "7/16/2010"  "7/7/2010"   "6/29/2010"  "6/10/2010"  "6/4/2010"   "5/27/2010"  "5/20/2010"  "5/10/2010"
 [10] "5/6/2010"   "2/4/2010"   "7/15/2009"  "7/2/2009"   "6/22/2009"  "5/18/2009"  "5/4/2009"   "4/20/2009"  "4/9/2009"
 [19] "3/30/2009"  "3/23/2009"  "3/17/2009"  "3/12/2009"  "3/10/2009"  "3/5/2009"   "3/2/2009"   "2/24/2009"  "2/23/2009"
 [28] "2/17/2009"  "2/10/2009"  "1/29/2009"  "1/28/2009"  "1/21/2009"  "1/20/2009"  "1/14/2009"  "1/7/2009"   "1/2/2009"
 [37] "12/16/2008" "12/8/2008"  "12/5/2008"  "12/4/2008"  "12/2/2008"  "12/1/2008"  "11/26/2008" "11/24/2008" "11/21/2008"
 [46] "11/20/2008" "11/19/2008" "11/14/2008" "11/13/2008" "11/12/2008" "11/6/2008"  "11/5/2008"  "11/4/2008"  "10/28/2008"
 [55] "10/27/2008" "10/24/2008" "10/22/2008" "10/21/2008" "10/20/2008" "10/16/2008" "10/15/2008" "10/13/2008" "10/9/2008"
 [64] "10/7/2008"  "10/6/2008"  "10/2/2008"  "9/30/2008"  "9/29/2008"  "9/22/2008"  "9/19/2008"  "9/18/2008"  "9/17/2008"
 [73] "9/15/2008"  "9/9/2008"   "9/4/2008"   "6/26/2008"  "6/6/2008"   "4/1/2008"   "3/18/2008"  "3/11/2008"  "2/5/2008"
 [82] "1/17/2008"  "11/7/2007"  "8/9/2007"   "2/27/2007"  "3/24/2003"  "3/17/2003"  "3/13/2003"  "1/24/2003"  "1/2/2003"
 [91] "10/15/2002" "10/11/2002" "10/10/2002" "10/1/2002"  "9/27/2002"  "9/19/2002"  "9/3/2002"   "8/14/2002"  "8/8/2002"
[100] "8/6/2002"   "8/5/2002"   "8/1/2002"   "7/29/2002"  "7/24/2002"  "7/22/2002"  "7/19/2002"  "7/10/2002"  "7/5/2002"
[109] "5/8/2002"   "9/24/2001"  "9/20/2001"  "9/17/2001"  "4/18/2001"  "4/5/2001"   "4/3/2001"   "3/12/2001"  "1/3/2001"
```



```
> count
[1] 216
> outlier_dates
  [1] "9/1/2010"   "7/16/2010"  "7/7/2010"   "6/29/2010"  "6/10/2010"  "6/4/2010"   "5/27/2010"  "5/20/2010"  "5/10/2010"
 [10] "5/6/2010"   "2/4/2010"   "7/15/2009"  "7/2/2009"   "6/22/2009"  "5/18/2009"  "5/4/2009"   "4/20/2009"  "4/9/2009"
 [19] "3/30/2009"  "3/23/2009"  "3/17/2009"  "3/12/2009"  "3/10/2009"  "3/5/2009"   "3/2/2009"   "2/24/2009"  "2/23/2009"
 [28] "2/17/2009"  "2/10/2009"  "1/29/2009"  "1/28/2009"  "1/21/2009"  "1/20/2009"  "1/14/2009"  "1/7/2009"   "1/2/2009"
 [37] "12/16/2008" "12/8/2008"  "12/5/2008"  "12/4/2008"  "12/2/2008"  "12/1/2008"  "11/26/2008" "11/24/2008" "11/21/2008"
 [46] "11/20/2008" "11/19/2008" "11/14/2008" "11/13/2008" "11/12/2008" "11/6/2008"  "11/5/2008"  "11/4/2008"  "10/28/2008"
 [55] "10/27/2008" "10/24/2008" "10/22/2008" "10/21/2008" "10/20/2008" "10/16/2008" "10/15/2008" "10/13/2008" "10/9/2008"
 [64] "10/7/2008"  "10/6/2008"  "10/2/2008"  "9/30/2008"  "9/29/2008"  "9/22/2008"  "9/19/2008"  "9/18/2008"  "9/17/2008"
 [73] "9/15/2008"  "9/9/2008"   "9/4/2008"   "6/26/2008"  "6/6/2008"   "4/1/2008"   "3/18/2008"  "3/11/2008"  "2/5/2008"
 [82] "1/17/2008"  "11/7/2007"  "8/9/2007"   "2/27/2007"  "3/24/2003"  "3/17/2003"  "3/13/2003"  "1/24/2003"  "1/2/2003"
 [91] "10/15/2002" "10/11/2002" "10/10/2002" "10/1/2002"  "9/27/2002"  "9/19/2002"  "9/3/2002"   "8/14/2002"  "8/8/2002"
[100] "8/6/2002"   "8/5/2002"   "8/1/2002"   "7/29/2002"  "7/24/2002"  "7/22/2002"  "7/19/2002"  "7/10/2002"  "7/5/2002"
[109] "5/8/2002"   "9/24/2001"  "9/20/2001"  "9/17/2001"  "4/18/2001"  "4/5/2001"   "4/3/2001"   "3/12/2001"  "1/3/2001"
[118] "12/20/2000" "12/5/2000"  "10/19/2000" "10/13/2000" "5/30/2000"  "4/25/2000"  "4/17/2000"  "4/14/2000"  "3/16/2000"
[127] "2/18/2000"  "1/4/2000"   "10/28/1999" "10/15/1998" "10/1/1998"  "9/30/1998"  "9/23/1998"  "9/11/1998"  "9/8/1998"
[136] "9/1/1998"   "8/31/1998"  "8/27/1998"  "8/4/1998"   "1/9/1998"   "10/28/1997" "10/27/1997" "9/2/1997"   "3/8/1996"
[145] "11/15/1991" "8/21/1991"  "1/17/1991"  "8/27/1990"  "8/23/1990"  "8/6/1990"   "10/13/1989" "5/31/1988"  "4/14/1988"
[154] "1/8/1988"   "1/4/1988"   "12/3/1987"  "11/30/1987" "11/9/1987"  "10/29/1987" "10/26/1987" "10/22/1987" "10/21/1987"
[163] "10/20/1987" "10/19/1987" "10/16/1987" "10/14/1987" "9/11/1986"  "7/7/1986"   "11/30/1982" "11/3/1982"  "10/25/1982"
[172] "10/6/1982"  "8/20/1982"  "8/17/1982"  "8/24/1981"  "4/22/1980"  "3/24/1980"  "3/17/1980"  "10/9/1979"  "11/1/1978"
[181] "1/27/1975"  "11/18/1974" "10/29/1974" "10/23/1974" "10/9/1974"  "10/7/1974"  "9/19/1974"  "9/5/1974"   "8/30/1974"
[190] "7/12/1974"  "7/8/1974"   "12/26/1973" "11/26/1973" "11/19/1973" "5/24/1973"  "8/16/1971"  "5/27/1970"  "11/26/1963"
[199] "10/24/1962" "6/28/1962"  "6/4/1962"   "5/29/1962"  "5/28/1962"  "4/18/1961"  "4/17/1961"  "10/23/1957" "10/21/1957"
[208] "10/10/1955" "9/26/1955"  "7/6/1955"   "6/6/1955"   "2/9/1953"   "12/4/1950"  "11/28/1950" "6/29/1950"  "6/26/1950"
```