CMDA 3654: Assignment #1

Due on Friday, Sep 9, 2016

Jung Choi

"I have neither given nor received unauthorized assistance on this assignment."

September 9, 2016

1 Life Spent Problem

(20points) Using R, compute the percentage of your life that you have spent at Virginia Tech. Report R code and output.

$$Pct = \frac{(currentyear - admissionyear)x12 + (currentmonth - admissionmonth)}{(currentyear - birthyear)x12 + (currentmonth - birthmonth)} x100$$

$$Pct = \frac{(2016 - 2012)x12 + (9 - 9)}{(2016 - 1992)x12 + (9 - 1)} x100 = 16.55$$

```
cuYear = 2016
cuMonth = 9
adYear = 2012
adMonth = 9
birthYear = 1992
birthMonth = 1

pct = ((cuYear-adYear)*12+(cuMonth-adMonth))/
          ((cuYear-birthYear)*12+(cuMonth-birthMonth))
pct = pct * 100
print(pct)
[1] 16.55405
```

2 Fibonacci Problem

(20points) Using R, compute the first 1000 Fibonacci numbers starting with 1, and plot the ratios for n = 1,2,...,999. Report R code and R plot.

```
# list all numbers from 1 to 999
x1 = 0 #  X0 = 0
x2 = 1 #  X-1 = -1
x=NULL
y=NULL
for(i in 1:999){                  # set to param 1:999
  x = c(x, x1 + x2)                # Xn = Xn-1 + Xn-2
  y = c(y, (x1+x2)+x1)             # Xn+1 = Xn + Xn-1
  x2 = x1                          # Update
  x1 = x[i]
}
plot(y/x, type="p", main = "Xn+1/Xn ratio", xlab="n", ylab="Xn+1/Xn" )
```

3 Iris data set problem

(20points) From UC Irvine's machine learning depository, consider the Iris data set at http://archive.ics.uci.edu/ml/datasets/Iris.

(a) Using your own words, describe the data set in 2-3 sentences
There are two identical data sets(Iris and bezdekIris) provided in the link above where both data contain 3 classes of 50 instances for each. Each class will have four attributes measured

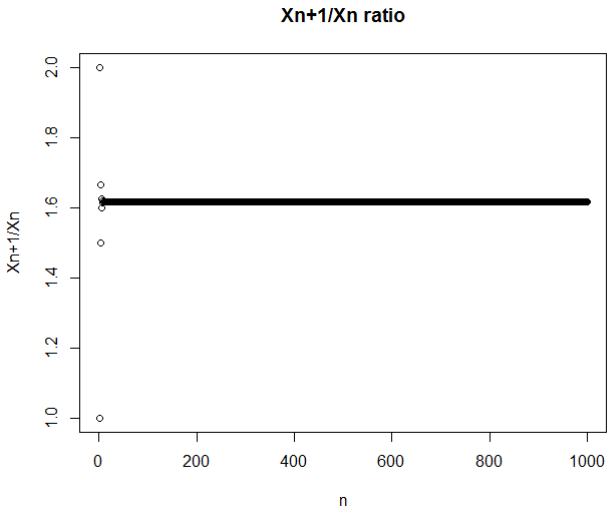

Figure 1: The ratio of the Xn+1/Xn using Fibonacci numbers

in cm and the last column indicates the class type. Both Iris and BezdekIris data sets are almost identical, so I picked one to demonstrate my approach for this problem.

(b) Import the data in R, and convert the data set into a data frame. Define column names for the data frame using the "Attribute Information" from the above web page.

```
# Commas are eliminated and col.names is used to create vectors for each
# column using the given attribute information. (i.e. data2$classes)
data2 = read.table("bezdekIris.data", sep = ",",
                   col.names = c("sepLen", "sepWid", "petLen", "petWid",
                                 "classes"))
```

(c) For each "class", calculate the mean and standard deviation of sepal length, sepal width, petal length,and petal width. Write this summary information into an R matrix called "summary". Report this summary matrix.

```
...

data2.classes = cbind(setosa = data2[1:50,1:5], versi = data2[51:100,1:5], virgi =
     data2[101:150,1:5])

# Commas are eliminated and col.names is used to create vectors for each
# column using the given attribute information. (i.e. data2$classes)
data2 = read.table("bezdekIris.data", sep = ",",
                   col.names = c("sepLen", "sepWid", "petLen", "petWid",
                                 "classes"))

data2.classes.setosaMean = c(mean(data2.classes$setosa.sepLen),
                             mean(data2.classes$setosa.sepWid),
                             mean(data2.classes$setosa.petLen),
```

```
                              mean(data2.classes$setosa.petWid))
data2.classes.versiMean = c(mean(data2.classes$versi.sepLen),
                            mean(data2.classes$versi.sepWid),
                            mean(data2.classes$versi.petLen),
                            mean(data2.classes$versi.petWid))
data2.classes.virgiMean = c(mean(data2.classes$virgi.sepLen),
                            mean(data2.classes$virgi.sepWid),
                            mean(data2.classes$virgi.petLen),
                            mean(data2.classes$virgi.petWid))

data2.classes.setosaSd = c(sd(data2.classes$setosa.sepLen),
                           sd(data2.classes$setosa.sepWid),
                           sd(data2.classes$setosa.petLen),
                           sd(data2.classes$setosa.petWid))
data2.classes.versiSd = c(sd(data2.classes$versi.sepLen),
                          sd(data2.classes$versi.sepWid),
                          sd(data2.classes$versi.petLen),
                          sd(data2.classes$versi.petWid))
data2.classes.virgiSd = c(sd(data2.classes$virgi.sepLen),
                          sd(data2.classes$virgi.sepWid),
                          sd(data2.classes$virgi.petLen),
                          sd(data2.classes$virgi.petWid))

summ = rbind(Setosa.Mean  = data2.classes.setosaMean,
             Versicolour.Mean = data2.classes.versiMean,
             Virginica.Mean = data2.classes.virgiMean,
             Setosa.Sd = data2.classes.setosaSd,
             Versicolour.Sd = data2.classes.versiSd,
             Virginica.Sd = data2.classes.virgiSd)


Measurement = c("sepal␣length␣in␣cm", "sepal␣width␣in␣cm",
                "petal␣length␣in␣cm", "petal␣width␣in␣cm")

# matrix form1
matrix(summ, byrow=T, 4, 6)

# matrix form2
summ = cbind(Setosa.Mean  = data2.classes.setosaMean,
             Versicolour.Mean = data2.classes.versiMean,
             Virginica.Mean = data2.classes.virgiMean,
             Setosa.Sd = data2.classes.setosaSd,
             Versicolour.Sd = data2.classes.versiSd,
             Virginica.Sd = data2.classes.virgiSd)
summary2 = data.frame(Measurement, summ)
summary2

...
```

(d) Save the iris data frame and the summary matrix in an R workspace called "iris.RData". Report R code and answers.

```
...

# Save the summary matrix and the iris data frame
save(summary2, data2, file="iris.RData")
rm(list=ls())
load("iris.RData")
```

	Measurement	Setosa Mean	Versicolour Mean	Virginica Mean	Setosa Sd	Versicolour Sd	Virginica Sd
1	sepal length in cm	5.006	5.936	6.588	0.3524897	0.5161711	0.6358796
2	sepal width in cm	3.428	2.770	2.974	0.3790644	0.3137983	0.3224966
3	petal length in cm	1.462	4.260	5.552	0.1736640	0.4699110	0.5518947
4	petal width in cm	0.246	1.326	2.026	0.1053856	0.1977527	0.2746501

4 "BABYNAMES"

(20points) Install the R package "babynames". Load the baby names data and answer the following questions. Report R code and answers.

(a) Describe the data set in two sentences. How many rows and columns does the data set have?

"babynames" is one of the data frame in the package called "babynames". The data frame contains five variables separated by each column labeled as year, sex, name, n, and prop. There are total number of 1825433 baby names which creates 1825433x5 data frame.

(b) How many unique names are there in the dataset? Why is this number different from the number of rows in (a)?

I figured the most optimal approach to find unique names is by first listing the data set's variable, n, increasing order. That way, we would see the names that are most uncommonly used or roughly we could say they are unique names. The data could contain one smallest varible n or there could be multiple numbers of names that have same n value. My data shows that the smallest n is 5. There are thousands of names in the data set where n = 5, so taking the whole data set as an object to be subsetted, we can then return a variable containing data set that is n <6.

```
...
library("babynames")

sub = subset(babynames, n < 6)
# unique()
sub2 = subset(sub, !duplicated(sub[,3]))
sub2
...
```

As a result, there are total number of 254,615 unique names. The data set from part (a) includes all babynames across all years where n ranges widely from 5 to 99680. Whereas "sub"

variable is a subset of babynames that contains data; n < 6.

(c) What were the most popular male names for the years 1900, 1925, 1950, 1975, 2000? What were the most popular female names for the years 2010, 2011, 2012, 2013, 2014?

```
# Using the %in% notation, I can subset the values of variable year that are
# equal to the given years for each gender.
subYearMale = babynames[babynames$year %in% c(1900, 1925, 1950, 1975, 2000), ]
subYearFemale = babynames[babynames$year %in% c(2010, 2011, 2012, 2013, 2014), ]

subMale = subset(subYearMale, sex == "M") # Separate the data set into male and
    female
subFemale = subset(subYearFemale, sex == "F")

# Find the max n for each year then match with the original data to creat a new
    subset.
popMale = rbind(subMale[subMale$n == max(subMale[subMale$year==1900, ][,4]), ],
                subMale[subMale$n == max(subMale[subMale$year==1925, ][,4]), ],
                subMale[subMale$n == max(subMale[subMale$year==1950, ][,4]), ],
                subMale[subMale$n == max(subMale[subMale$year==1975, ][,4]), ],
                subMale[subMale$n == max(subMale[subMale$year==2000, ][,4]), ])

popFemale = rbind(subFemale[subFemale$n == max(subFemale[subFemale$year==2010,
    ][,4]), ],
                  subFemale[subFemale$n == max(subFemale[subFemale$year==2011,
                      ][,4]), ],
                  subFemale[subFemale$n == max(subFemale[subFemale$year==2012,
                      ][,4]), ],
                  subFemale[subFemale$n == max(subFemale[subFemale$year==2013,
                      ][,4]), ],
                  subFemale[subFemale$n == max(subFemale[subFemale$year==2014,
                      ][,4]), ])
popMale
popFemale

> popMale
        year sex    name     n       prop
54491   1900   M    John  9829 0.06061709
229113  1925   M  Robert 60903 0.05289203
468034  1950   M   James 86221 0.04739490
784594  1975   M Michael 68457 0.04217672
1350191 2000   M   Jacob 34465 0.01651561
> popFemale
        year sex     name     n       prop
1657593 2010   F Isabella 22883 0.01169826
1691634 2011   F   Sophia 21816 0.01128975
1725503 2012   F   Sophia 22267 0.01152159
1759187 2013   F   Sophia 21147 0.01102227
1792390 2014   F     Emma 20799 0.01072924
>
```

5 10 MOST POPULAR BABIES

(20points) What are the 10 most popular male baby names across years? What are the 10 most popular female baby names across years?

```
popMaleAll = subset(babynames, sex == "M")  # Contains only male applicants
popFemaleAll = subset(babynames, sex == "F") # Contains only female applicants

popMaleAll2 = popMaleAll[order(popMaleAll$n, decreasing = TRUE), c(1:5)] # Sort n
    in descreasing order
popFemaleAll2 = popFemaleAll[order(popFemaleAll$n, decreasing =TRUE), c(1:5)]

# !duplicated(popMaleAll[,3])
# Remove duplicated names and sort n in order which has to be done
# first so that we have correct output from most popular to least popular names
popMaleAll2 = subset(popMaleAll2, !duplicated(popMaleAll2[,3]))
popFemaleAll2 = subset(popFemaleAll2, !duplicated(popFemaleAll2[,3]))

popMaleAll2[1:10,]
popFemaleAll2[1:10,]

> popMaleAll2[1:10,]
       year sex        name     n       prop
437158 1947   M       James 94755 0.05101816
544603 1957   M     Michael 92709 0.04238006
437159 1947   M      Robert 91642 0.04934205
437160 1947   M        John 88318 0.04755233
521860 1955   M       David 86191 0.04126896
437161 1947   M     William 66969 0.03605757
953887 1984   M Christopher 60016 0.03199918
427041 1946   M     Richard 58859 0.03567093
579780 1960   M        Mark 58735 0.02711617
819638 1977   M       Jason 55649 0.03254811

> popFemaleAll2[1:10,]
        year sex     name     n       prop
431053  1947   F    Linda 99680 0.05483648
180216  1921   F     Mary 73985 0.05781614
726661  1972   F Jennifer 63606 0.03944644
633420  1965   F     Lisa 60268 0.03298245
472231  1951   F Patricia 56422 0.03055766
1001933 1987   F  Jessica 55985 0.02988289
1001934 1987   F   Ashley 54840 0.02927173
504178  1954   F  Deborah 54677 0.02746644
515140  1955   F    Debra 50540 0.02521545
431056  1947   F  Barbara 48793 0.02684226
```