Activity 2/3 - Gioconda Prada

Note: in this document, you will see that math is written funny. The symbols (like dollar signs and backslashes) are the way to get math looking nice in this sort of document. You will not need to be writing any equations, so you don't need to learn it. Often it can be hard reading what the equation says though, so I recommend Knitting the document to HTML or to Word to see the equation more clearly.

Directions: make sure this file is in your STA 9750 or OPR 9750 folder (along with the data file you are asked to download in Question 3). Often when you download things they will go to your Downloads folder so make sure you have moved it. R will give an error if it tries to read in a file that is not in the same folder as the .Rmd file you are working from.

Put your code on separate lines between the back quotes. *Do not delete the backquotes*. This is how R knows which lines to run while making the HTML/Word document.

If a question has *Comment:*, this is a place for you to respond to the question being asked. On activities and homework, if you see *Comment:* or *Response:*, be sure to answer (if a question lacks such a spot, no response is required).

Activity 2

Question 2: Use the `seq` command to help answer the following questions:

a. What's the 111st element of a sequence that starts at 1.2 and increments by .3? (e.g., 1.2, 1.5, 1.8, ...)

b. Consider a sequence that starts at 5 and ends at 8 and which has 321 elements. How many elements are greater than 6.3?

```
#a
x <- seq(from=1.2,by=0.3,length=111)
x[111]
```

```
## [1] 34.2
```

```r
#b
x <- seq(from=5,to=8,length=321)
subset(x, x>6.3)
```

```
##   [1] 6.303125 6.312500 6.321875 6.331250 6.340625 6.350000 6.359375
6.368750
##   [9] 6.378125 6.387500 6.396875 6.406250 6.415625 6.425000 6.434375
6.443750
##  [17] 6.453125 6.462500 6.471875 6.481250 6.490625 6.500000 6.509375
6.518750
##  [25] 6.528125 6.537500 6.546875 6.556250 6.565625 6.575000 6.584375
6.593750
##  [33] 6.603125 6.612500 6.621875 6.631250 6.640625 6.650000 6.659375
6.668750
##  [41] 6.678125 6.687500 6.696875 6.706250 6.715625 6.725000 6.734375
6.743750
##  [49] 6.753125 6.762500 6.771875 6.781250 6.790625 6.800000 6.809375
6.818750
##  [57] 6.828125 6.837500 6.846875 6.856250 6.865625 6.875000 6.884375
6.893750
##  [65] 6.903125 6.912500 6.921875 6.931250 6.940625 6.950000 6.959375
6.968750
##  [73] 6.978125 6.987500 6.996875 7.006250 7.015625 7.025000 7.034375
7.043750
##  [81] 7.053125 7.062500 7.071875 7.081250 7.090625 7.100000 7.109375
7.118750
##  [89] 7.128125 7.137500 7.146875 7.156250 7.165625 7.175000 7.184375
7.193750
##  [97] 7.203125 7.212500 7.221875 7.231250 7.240625 7.250000 7.259375
7.268750
## [105] 7.278125 7.287500 7.296875 7.306250 7.315625 7.325000 7.334375
7.343750
## [113] 7.353125 7.362500 7.371875 7.381250 7.390625 7.400000 7.409375
7.418750
## [121] 7.428125 7.437500 7.446875 7.456250 7.465625 7.475000 7.484375
7.493750
## [129] 7.503125 7.512500 7.521875 7.531250 7.540625 7.550000 7.559375
7.568750
## [137] 7.578125 7.587500 7.596875 7.606250 7.615625 7.625000 7.634375
7.643750
## [145] 7.653125 7.662500 7.671875 7.681250 7.690625 7.700000 7.709375
7.718750
## [153] 7.728125 7.737500 7.746875 7.756250 7.765625 7.775000 7.784375
7.793750
## [161] 7.803125 7.812500 7.821875 7.831250 7.840625 7.850000 7.859375
7.868750
## [169] 7.878125 7.887500 7.896875 7.906250 7.915625 7.925000 7.934375
7.943750
## [177] 7.953125 7.962500 7.971875 7.981250 7.990625 8.000000
```

4. Consider various columns of ALUMNI. Write all code to answer the following questions in the single R chunk that's after the list of questions.

b. What *percentage* (a number between 0-1) of alumni have NOT donated. Two ways of approaching this: one with `length` and `which` and the other with `mean` (we recommend doing both!). One way is to count how many the entries in the LTG2UT column equal 0 and divide by the number of rows of the data.

Response:

c. "Print to the screen" the values of GENDER_CODE in rows 201, 320, 474, 479, and 533. Remember the phrase "print to the screen" just means ensuring the command does not have a left-arrow so that the results get output to the console and are "knitted" into the document.

d. Which row (1-43994) contains the largest value of LTG2UT?

Response:

g. How many rows have values LTG2UT that are greater than 500,000 dollars AND values of AREA_CODE that is 865?

Response:

j. Print to the screen the possible values (i.e., levels vector) of DEGREE1_CAMPUS

Response:

```r
load("Activity2.RData")

#b
x<-length(which(ALUMNI$LTG2UT==0))
x/nrow(ALUMNI)

## [1] 0.588353

#c
rows<- c(201, 320, 474, 479, 533)
print(ALUMNI$GENDER_CODE[rows])

## [1] M M M M F
## Levels: F M U

#f
max(ALUMNI$LTG2UT)

## [1] 48879436

#g
sum(ALUMNI$LTG2UT>500000 & ALUMNI$AREA_CODE==865, na.rm = TRUE)
```

```
## [1] 23

#j
print (levels(ALUMNI$DEGREE1_CAMPUS))

## [1] " "                 "HSC"               "UT Medical Center"
## [4] "UTC"               "UTK"               "UTM"
## [7] "UTN"
```

---

5. Rename the GENDER_CODE column to be GENDER, rename its levels to be "Female", "Male", and "Unknown", and print to the screen a frequency table of this recoded GENDER column.

```
ALUMNI$GENDER <- ALUMNI$GENDER_CODE
levels(ALUMNI$GENDER) <- c("Female", "Male", "Unknown")
table(ALUMNI$GENDER)

##
##  Female    Male Unknown
##   14427   29484      83
```

Activity 3

Question 1: Download Act3-CustomerTrans.dat and put it in your STA 9750 or OPR 9750 folder. This is the entirety of some customer's credit card transaction data. Each record has the amount of the transaction, the merchant involved (coded by, M326, for example, UNKNOWN when not available), the type of merchant (industry and super industry), the merchant's and customer's zip codes when available (along with the distance between these zips), whether the purchase was made on the internet, and the date of the purchase (measured in # of days since the beginning of data collection).

a. Import the file and name the dataframe CUST. Try this using read.csv. Determine the number of rows and columns in the dataset by running nrow(), ncol(), or dim() (using CUST as the argument to these functions), then show the first four lines of the data using head (see RBasic slides).

```
CUST <- read.csv("Act3-CustomerTrans.dat")
nrow(CUST)

## [1] 1551

ncol(CUST)

## [1] 10

dim(CUST)

## [1] 1551   10

head(CUST,4)
```

```
##   Amount DaysSinceStart                Industry            SuperIndustry
## 1  14.06             33           Family Apparel  Apparel and Accessories
## 2   4.99            409           Grocery Stores  Grocery and Food Stores
## 3  19.53            311 Home Improvement Centers Home Improvement Centers
## 4   0.99            135            Miscellaneous            Miscellaneous
##   Internet  CZIP  MZIP    ID Merchant Distance
## 1       No 10583 10917 37783      M20       26
## 2       No 10583 10549 37783      M29       13
## 3       No 10583 11710 37783      M65       22
## 4      Yes 10583 98144 37783     M155     2092
```

---

b. Show row 27 of the data on the screen. What is the amount spent for this transaction?

```
CUST [27,]

##    Amount DaysSinceStart  Industry         SuperIndustry Internet  CZIP
MZIP
## 27     95            362 Insurance Finance and Insurance       No 10583
51401
##       ID Merchant Distance
## 27 37783     M179      949
```

Comment:

---

d. Using either `summary` or `table`, find how many times this person purchased something on the internet, column `Internet`.

```
internet_table<- table(CUST$Internet)
print(internet_table)

##
##   No  Yes
## 1405  146
```

Comment:
