Task 1 - Assignment 1

Frederick Abi Chahine

9/29/2021

Exercise - 1

```
clients_data = read.csv("Clients.csv")
```

The collected information on each client consists of:

```
colnames(clients_data)
```

```
##  [1] "clientID"              "gender"                "maritalStatus"
##  [4] "works"                 "income"                "healthCoverage"
##  [7] "housingStatus"         "recentlyChangedHousing" "numberCars"
## [10] "age"                   "state"
```

- **clientID:** Numerical attribute that contains unique IDs for each client.
- **gender:** Categorical attribute with *F* representing *Females* & *M* representing *Males*.
- **maritalStatus:** Categorical attribute that indicates whether a client is *Married*, *Never Married*, or *DivorcedSeperated*.
- **works:** Binomial attribute that shows *True* if a client works, *False* if not.
- **income:** Numerical attribute that holds the income of each client.
- **healthCoverage:** Binomial attribute that shows *True* if a client has health coverage, *False* if not.
- **housingStatus:** Categorical attribute that shows if a client is a homeowner free and clear, homeowner with mortgageloan, occupied a house with no rent, or rented a house.
- **recentlyChangedHousing:** Binomial attribute that shows *True* if a client recently changed houses, *False* if not.
- **numberCars:** Numerical attribute that displays the number of cars a client has.
- **age:** Numerical attribute that states the age of each client.
- **state:** Categorical attribute that shows the state each client is from.

Exercise - 2

```
summary(clients_data)
```

```
##     clientID          gender          maritalStatus        works
##  Min.   :    1   Length:1000        Length:1000        Mode :logical
##  1st Qu.: 347385   Class :character   Class :character   FALSE:73
##  Median : 695121   Mode  :character   Mode  :character   TRUE :599
##  Mean   : 700218                                       NA's :328
##  3rd Qu.:1044224
##  Max.   :1416004
##
##      income        healthCoverage  housingStatus       recentlyChangedHousing
##  Min.   : -8700   Mode :logical   Length:1000        Mode :logical
##  1st Qu.: 14600   FALSE:159       Class :character   FALSE:829
##  Median : 35000   TRUE :841       Mode  :character   TRUE :124
##  Mean   : 53505                                      NA's :56
##  3rd Qu.: 67000
##  Max.   :615000
##
##    numberCars          age            state
##  Min.   :0.000   Min.   :  0.0   Length:1000
##  1st Qu.:1.000   1st Qu.: 38.0   Class :character
##  Median :2.000   Median : 50.0   Mode  :character
##  Mean   :1.516   Mean   : 51.7
##  3rd Qu.:2.000   3rd Qu.: 64.0
##  Max.   :6.000   Max.   :147.0
##  NA's   :56
```

```
classes=sapply(clients_data , class)

character_classes=which(classes=="character")

for(i in character_classes) {
  clients_data[,i]=as.factor(clients_data[,i])
}
```

From the summary of the whole data, we can see that some categorical variables were read into R as *character* attributes.
The code above changes their class back into factor.

```
summary(clients_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    38.0    50.0    51.7    64.0   147.0
```

From the summary of the age attribute, we can note that:

- *the minimum value is 0.0*
- *the maximum value is 147.0*
- *the median value is 50.0*
- *the average / mean value is 51.7*

```
summary(clients_data$income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -8700   14600   35000   53505   67000  615000
```

From the summary of the income attribute, we can note that:

- *the minimum value is -8700*
- *the maximum value is 615000*
- *the median value is 35000*
- *the average / mean value is 53505*

```
summary(clients_data$housingStatus)
```

```
##    Homeowner free and clear Homeowner with mortgageloan
##                         157                          412
##         Occupied with no rent                      Rented
##                          11                          364
##                        NA's
##                          56
```

From the summary of the *housingStatus* attribute, we can note that:

- *there are 157 homeowner's free and clear*
- *there are 412 homeowner's with mortgageloan*
- *there are 11 occupied with no rent*
- *there are 364 rented*
- *there are 56 missing values*

Exercise - 3

```
temp_clients_data = clients_data
age_income_index = which((temp_clients_data$age > 0) & (temp_clients_data$age < 100) & (temp_clients_data$income
 > 0))
temp_clients_data = temp_clients_data[age_income_index,]
cor.test(temp_clients_data$age, temp_clients_data$income)
```

```
##
##  Pearson's product-moment correlation
##
## data:  temp_clients_data$age and temp_clients_data$income
## t = -0.6754, df = 908, p-value = 0.4996
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08729917  0.04254146
## sample estimates:
##         cor
## -0.02240845
```

There is no correlation. As seen from the cor.test() function, the cor value (-0.022) is not significant enough to indicate any form of correlation. Additionally, the p-value is larger than 0.05 (5%) which is not statistically significant and indicates no correlation.

```
plot(temp_clients_data$age, temp_clients_data$income, xlab = "Age", ylab = "Income", main = "Scatter Plot of Age
 vs Income",pch=19,col="blue" )
```

The scatter plot above displays a small trend in which income slightly increases for the middle aged group ~ roughly ages 35 -70 (a small parabolic/quadratic relationship); However, this does not seem to be significant enough for there to be a set correlation.

Exercise - 4

```
summary(clients_data$housingStatus)
```

```
##    Homeowner free and clear Homeowner with mortgageloan
##                         157                          412
##         Occupied with no rent                      Rented
##                          11                          364
##                        NA's
##                          56
```

From the summary, we can deduce that there are 56 missing values in housingStatus.
This equates to 5.6% of the total observations.

```
summary(clients_data$recentlyChangedHousing)
```

```
##    Mode   FALSE    TRUE    NA's
## logical     829     124      56
```

From the summary, we can deduce that there are 56 missing values in recentlyChangedHousing.
This equates to 5.6% of the total observations.

```
summary(clients_data$numberCars)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.516   2.000   6.000      56
```

From the summary, we can deduce that there are 56 missing values in numberCars.
This equates to 5.6% of the total observations.

```
check_1 = which(is.na(clients_data$housingStatus)) == which(is.na(clients_data$recentlyChangedHousing))
check_2 = which(is.na(clients_data$recentlyChangedHousing)) == which(is.na(clients_data$numberCars))
check_1
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [14] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [27] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [40] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
check_2
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [14] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [27] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [40] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

In the line of code above, I am checking which indices have missing values in *housingStatus*, those missing in *recentlyChangedHousing*, & those missing in *numberCars*; Then, I am checking if these set of indices are equal or not.
From the results, we can see that those missing values come from the same observations.

In my opinion, we can deal with them by replacing the missing values with either the mean for the numerical attribute or the mode for the categorical attributes since the total number of missing values in these variables are only 5.6%.

```
#clients_data$housingStatus[which(is.na(clients_data$housingStatus))] <- mode
#clients_data$recentlyChangedHousing[which(is.na(clients_data$recentlyChangedHousing))] <- mode
#clients_data$numberCars[which(is.na(clients_data$numberCars))] <- mean

#clients_data$numberCars[which(is.na(clients_data$numberCars))] <- ceiling(mean(clients_data$numberCars[-which(is.
na(clients_data$numberCars))]))
```

Exercise - 5

```
summary(clients_data$works)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      73     599     328
```

The number of missing values is quite high as they equate to 32.8% of the total observations.
It does NOT make sense to remove all the observations with missing values since they represent more than 25% of the entire data set.

```
clients_data[,12]=clients_data$works

names(clients_data)[12] <- "fixedWorks"

clients_data$fixedWorks = as.factor(clients_data$fixedWorks)
temp = clients_data$fixedWorks

temp = ifelse(temp=="TRUE", "employed", "not employed")

temp[which(is.na(temp))]="missing"
clients_data$fixedWorks=temp
```

In the chunk of code above, we are defining a new variable **fixedWorks** that is a derivative of **works** in which all *TRUE* observations in *works* are replaced with *employed* in *fixedWorks*, all *FALSE* observations in *works* are replaced with *not employed* in *fixedWorks*, and finally all *NA* observations in *works* are replaced with *missing* in *fixedWorks*.
Of course, we initially had to change *fixedWorks* from the logical class into the factor class in order to perform these functions properly.

Exercise - 6

```
class(clients_data$income)
```

```
## [1] "integer"
```

```
missing = which(clients_data$income == 0)
length(missing)
```

```
## [1] 78
```

As we can see from the output, the type of variable *income* is an integer & the number of missing values (assume that a missing value is a value = 0) in this variable is 78.

```
clients_data[,13]=clients_data$income
names(clients_data)[13] <- "fixedIncome"
temp = clients_data$fixedIncome
mean_value = mean(temp[-missing])
temp = ifelse(temp==0, mean_value, temp)
clients_data$fixedIncome = temp
```

In the chunk of code above, we are defining a new variable **fixedIncome** that is a derivative of **income** in which all values that are equal to 0 (NA) in *income* are replaced by the mean in the new variable *fixedIncome*.

Exercise - 7

I do not think that having age as a *numerical* attribute helps us much as we do not care about the actual numerical value of the age attribute. We care more about how many clients fall within an age range; Therefore, it will be better to have age as a range (categorical) rather than having the actual values (numerical) to predict health coverage.

```
clients_data[,14] = cut(clients_data$age, breaks = c(0, 25, 65, Inf), labels = c("0-25", "26-65", "66-Inf"), incl
ude.lowest = TRUE)
names(clients_data)[14] <- "age.range"
```

The code above is simply utilizing the cut function in order to create a new variable *age.range* that is converting the variable from continuous to discrete since age is now a range rather than a single number.