Exercise 1B

Hanzhong Shen

2016/10/22

Exercise 1A

Preparation: Load the libraries

```
library(Hmisc)
library(dplyr)
```

Exercise 1.1: Loading data into R

1. Load the demo data set Loan_Data.csv into a data.frame using an appropriate R function.

```
demo <- read.csv("Loan_Data.csv", header = T, sep = ";")
```

2. Check the data type of the columns PHON and BAD in the data set. Is this data type appropriate? If not, convert the variables into a more suitable data type.

```
demo$PHON <- as.factor(demo$PHON)
demo$BAD <- as.factor(demo$BAD)
```

3. Make sure that values of PHON and BAD are, respectively, yes/no and good/bad.

```
levels(demo$PHON) <- c("no", "yes")
levels(demo$BAD) <- c("no", "yes")
```

4. Summarize the distribution of all columns in the data set using an appropriate R function. Remember that you can easily find appropriate functions and code snippets via the R help or Google. Every programmer constantly does this, so you should, too.

```
demo$nDEP <- as.factor(demo$nDEP)
summary(demo)
```

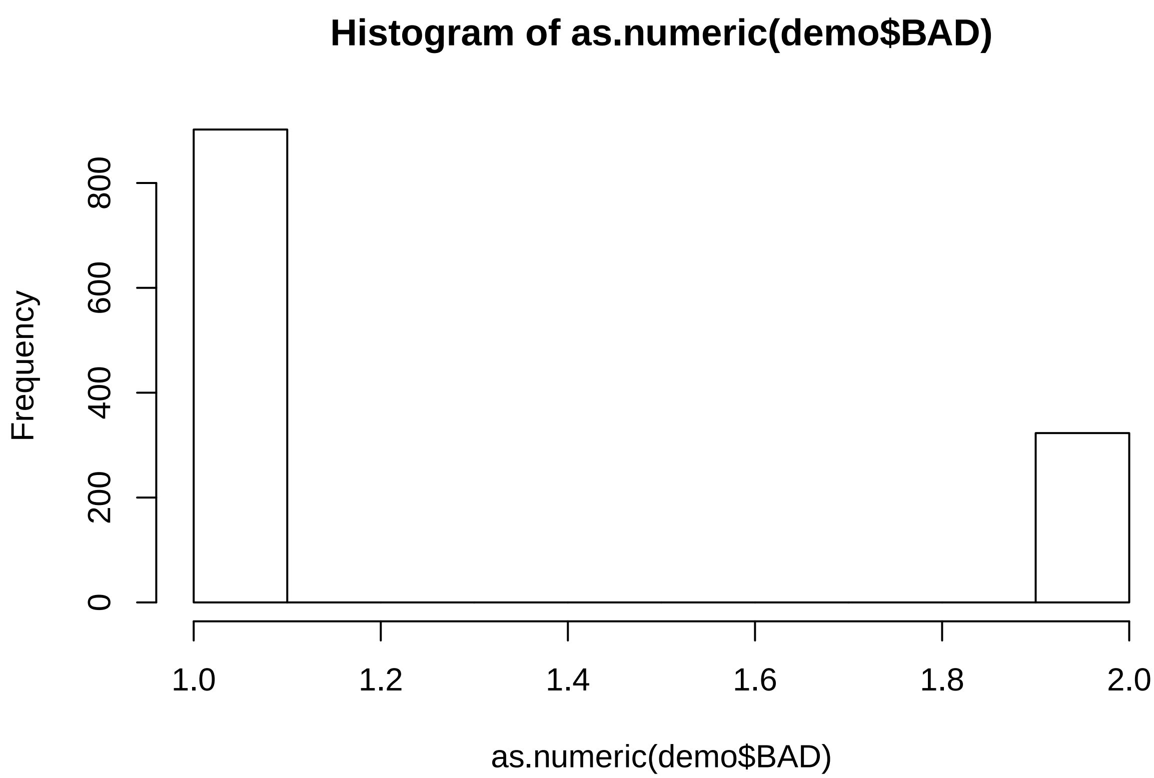
```
##       YOB            nKIDS        nDEP      PHON        dINC_SP
##  Min.   : 3.00   Min.   :0.0000   0:1185   no : 118   Min.   :    0
##  1st Qu.:42.00   1st Qu.:0.0000   1:  33   yes:1107   1st Qu.:    0
##  Median :55.00   Median :0.0000   2:   7              Median :    0
##  Mean   :51.04   Mean   :0.6237                       Mean   : 1990
##  3rd Qu.:63.00   3rd Qu.:1.0000                       3rd Qu.: 1040
##  Max.   :99.00   Max.   :5.0000                       Max.   :50000
##
##      EMPS_A        dINC_A       RES         dHVAL           dMBO
##  P      :531   Min.   :    0   F:129   Min.   :    0   Min.   :    0
##  V      :231   1st Qu.: 9000   N: 66   1st Qu.:    0   1st Qu.:    0
##  E      :124   Median :19500   O:624   Median :    0   Median :    0
##  T      :123   Mean   :21244   P:252   Mean   :15694   Mean   :11226
```

```
##  R       :104   3rd Qu.:30600   U:154   3rd Qu.:28928   3rd Qu.:20000
##  W       : 37   Max.   :64800           Max.   :64928   Max.   :64000
##  (Other): 75
##      dOUTM          dOUTL            dOUTHP          dOUTCC
##  Min.   :   0   Min.   :    0.0   Min.   :   0.00   Min.   :   0.0
##  1st Qu.:   0   1st Qu.:    0.0   1st Qu.:   0.00   1st Qu.:   0.0
##  Median : 256   Median :    0.0   Median :   0.00   Median :   0.0
##  Mean   : 342   Mean   :  121.9   Mean   :  28.72   Mean   :  39.6
##  3rd Qu.: 528   3rd Qu.:    0.0   3rd Qu.:   0.00   3rd Qu.:   0.0
##  Max.   :3800   Max.   :28000.0   Max.   :1600.00   Max.   :2800.0
##
##   BAD
##  no :902
##  yes:323
##
##
##
##
##
```

Exercise 1.2: Plotting a histogram

1. Create a histogram plot that depicts the distribution of BAD. Don't be surprised if your first attempt fails. This only shows that your solution to task 2 in exercise 1 was correct. Strangely, the hist() function requires numeric data. We could find a better function to do the plotting. Yet, in this exercise, let's do something else: Without changing your data.frame, simply convert BAD back to a numeric variable in the call of the hist() function.

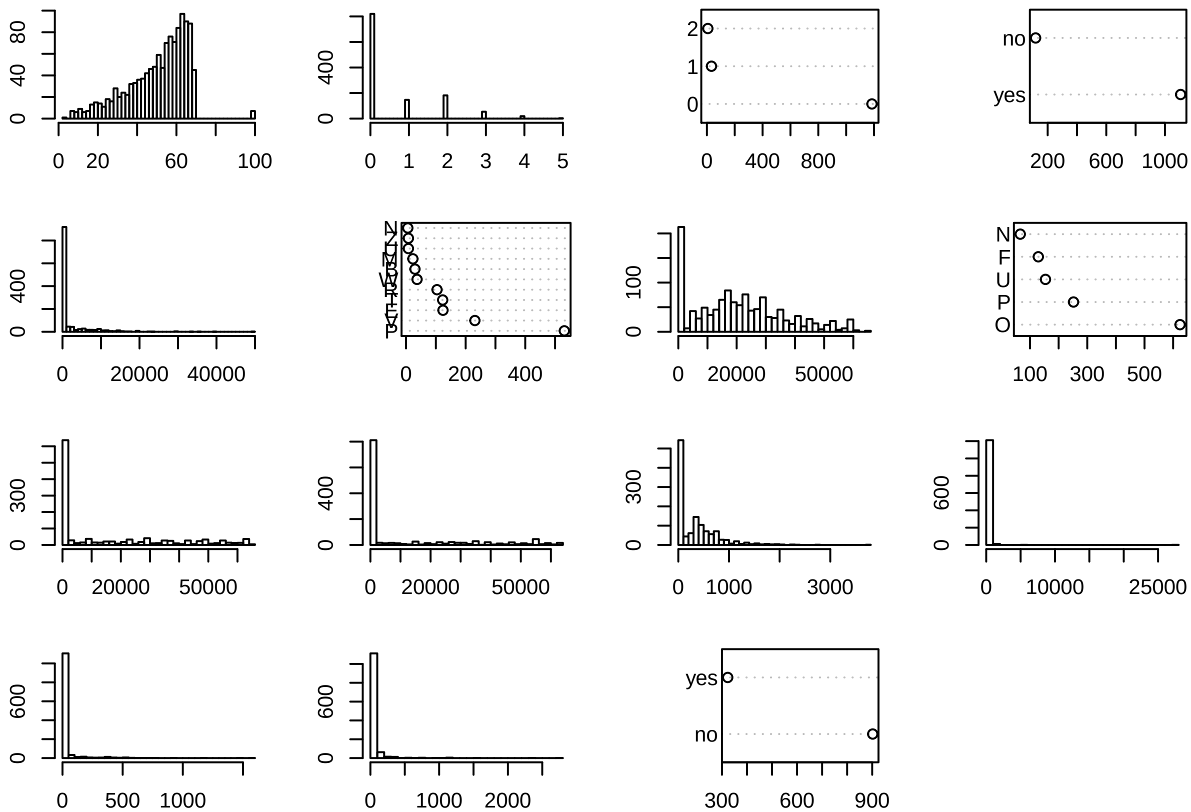
```r
hist(as.numeric(demo$BAD))
```



2. The hist() function gives us a way to plot the distribution of one variable in a data.frame. There is an alternative function that allows you to create a histogram plot for all variables in a data.frame. Which function is that? Hint: You will need to install package Hmisc.

3. Use the function identified in task 2 to depict the distribution of all variables in a matrix of histograms.

```
par(mar = rep(2, 4))
hist.data.frame(demo)
```



Exercise 1.3: Statistical analysis

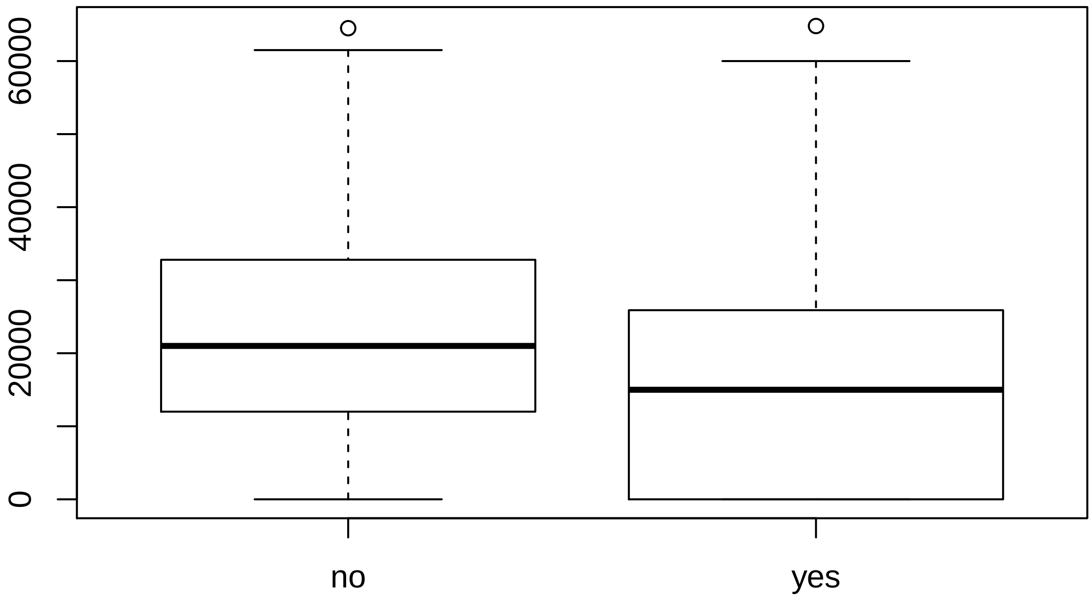
Recall that the column BAD gives the risk status (good or bad) for each applicant. Applicants' income is available in the column dINC_A. To separate the good and the bad risks in the data.frame, you can use logical indexing (with variable[index] and matrix[row index, column index]).

1. Create two variables, inc.good and inc.bad, which contain the incomes of good and bad credit risks, respectively. You will need to use logical indexing to select the correct rows and data.frame indexing to select the right column.

```
# inc.good <- filter(select(demo, dINC_A, BAD),BAD=='no')
# inc.bad <- filter(select(demo, dINC_A, BAD),BAD=='yes')
inc.good <- demo$dINC_A[demo$BAD == "no"]
inc.bad <- demo$dINC_A[demo$BAD == "yes"]
```

2. Depict the distribution of the income of customers with a good and bad risk, respectively, by means of a boxplot. Try Google or search the R help with ??boxplot. On average, which of the two groups earns more?

```r
boxplot(demo$dINC_A ~ demo$BAD)
```



From the plot, we can say that average income of customers with good risk is higher

3. Calculate the difference between the average/mean income of good and bad credit applicants.

```r
mean(inc.good)
```

```
## [1] 23008.64
```

```r
mean(inc.bad)
```

```
## [1] 16316.91
```

4. Identify an appropriate statistical test to verify whether the observed income difference is statistically significant. Perform the test and display its results. Hint: A Google search similar to "R statistical test difference in means" will help.

```r
result <- t.test(inc.good, inc.bad)
result
```

```
##
##  Welch Two Sample t-test
##
## data:  inc.good and inc.bad
## t = 6.7113, df = 585.43, p-value = 4.565e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4733.418 8650.041
## sample estimates:
## mean of x mean of y
##  23008.64  16316.91
```

5. Assign the test result to a variable. Use the print() function to output a message that tells the user whether the observed income difference is significant. You can do this with an if() condition by checking the the list entry p.value contained in the test result.

```
if (result$p.value <= 1 * 10^-3) {
    print("The observed income difference is Sigificant")
} else {
    "The observed income difference is not Sigificant"
}
```

```
## [1] "The observed income difference is Sigificant"
```