CDS501 PRINCIPLES & PRACTICES OF DATA SCIENCE & ANALYTICS

Lab 4

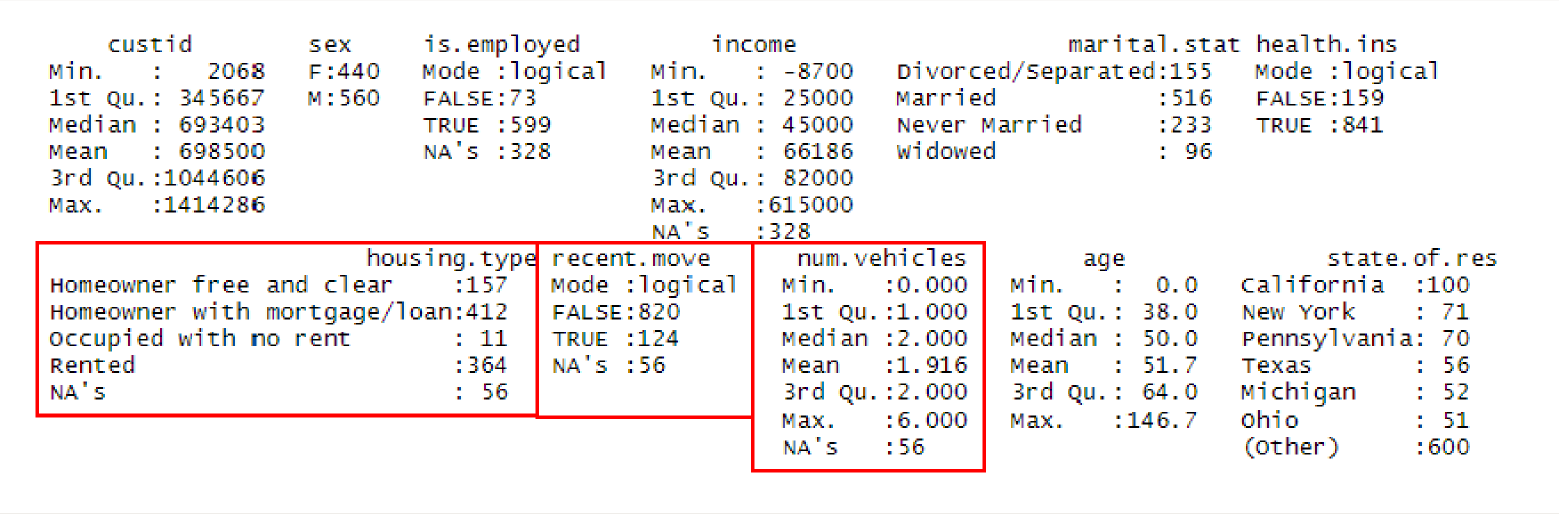
Managing Data with R

INTRODUCTION

- This tutorial attempts to demonstrate how to manage data with R. In this lab, we will be using customer dataset which can be downloaded from eLearn@USM
- Load the dataset into R, and name the data frame as custData.
- **custData <- read.table('cust.data.manage.csv', sep=',', header=T)**

HANDLING MISSING VALUES AND OUTLIERS

- **> summary(custData)** examine distribution of dataset.
- The command will show summary statistics on the numerical columns (attributes) and count statistics on the categorical columns



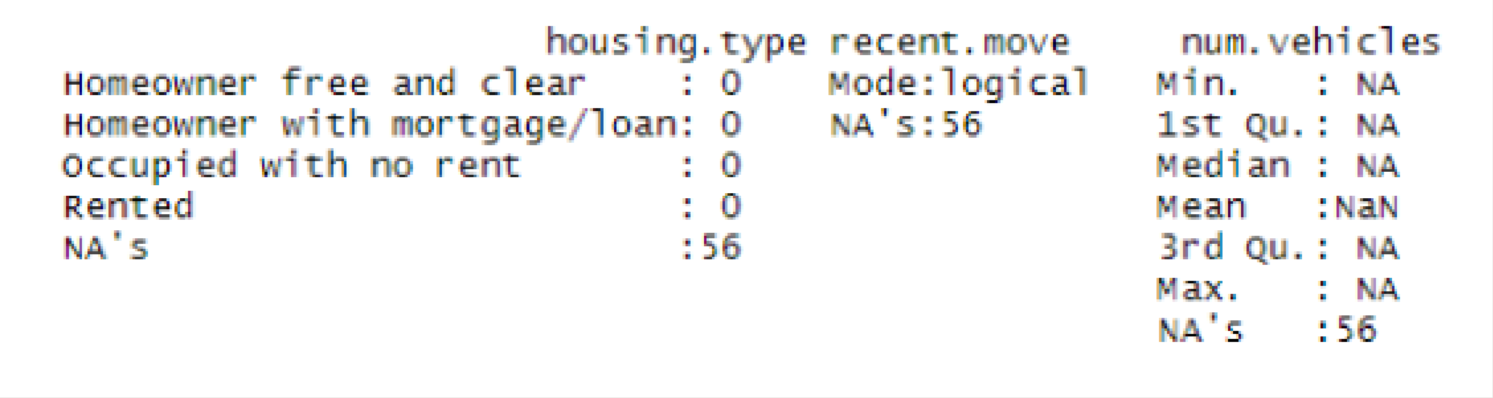
```
     custid          sex       is.employed        income                    marital.stat  health.ins
 Min.   :   2068   F:440   Mode :logical   Min.   : -8700   Divorced/Separated:155   Mode :logical
 1st Qu.: 345667   M:560   FALSE:73        1st Qu.: 25000   Married           :516   FALSE:159
 Median : 693403           TRUE :599       Median : 45000   Never Married     :233   TRUE :841
 Mean   : 698500           NA's :328       Mean   : 66186   Widowed           : 96
 3rd Qu.:1044606                           3rd Qu.: 82000
 Max.   :1414286                           Max.   :615000
                                           NA's   :328
                       housing.type recent.move     num.vehicles        age           state.of.res
 Homeowner free and clear    :157   Mode :logical   Min.   :0.000   Min.   :  0.0   California  :100
 Homeowner with mortgage/loan:412   FALSE:820       1st Qu.:1.000   1st Qu.: 38.0   New York    : 71
 Occupied with no rent       : 11   TRUE :124       Median :2.000   Median : 50.0   Pennsylvania: 70
 Rented                      :364   NA's :56        Mean   :1.916   Mean   : 51.7   Texas       : 56
 NA's                        : 56                   3rd Qu.:2.000   3rd Qu.: 64.0   Michigan    : 52
                                                    Max.   :6.000   Max.   :146.7   Ohio        : 51
                                                    NA's   :56                      (Other)     :600
```

DROPPING MISSING VALUES

- Let’s analyse the three attributes.
- **custData[is.na(custData$housing.type), c("housing.type", "recent.move", "num.vehicles")]**
- **summary(custData[is.na(custData$housing.type), c("housing.type", "recent.move", "num.vehicles")])**

DROPPING MISSING VALUES

- As we can see the three attributes missing exactly 56 values, means that it’s the same customers in each case. So, it’s probably safe to drop the rows with missing values



```
                      housing.type recent.move    num.vehicles
Homeowner free and clear    : 0    Mode:logical   Min.   : NA
Homeowner with mortgage/loan: 0    NA's:56        1st Qu.: NA
Occupied with no rent       : 0                   Median : NA
Rented                      : 0                   Mean   :NaN
NA's                        :56                   3rd Qu.: NA
                                                  Max.   : NA
                                                  NA's   :56
```

DROPPING MISSING VALUES

- To drop the rows, we create a subset of data frame without the rows with missing values.
- **custData_subset <- custData[!is.na(custData$housing.type),]**

FILLING MISSING VALUES IN CATEGORICAL DATA

- Customers might not in the active workforce and are not seeking paid employment.
- So, we group them into a single category. Here, we create a new category ("not in active workforce") and rename TRUE to "employed" and FALSE to "not employed".
- **custData_subset$is.employed.fix <- ifelse(is.na(custData_subset$is.employed), "not in active workforce", ifelse(custData_subset$is.employed==T, "employed", "not employed"))**

FILLING MISSING VALUES IN NUMERICAL DATA

- **.meanIncome <- mean(custData_subset$income, na.rm=T) -**Calculate the mean income
- **custData_subset$income.fix <- ifelse(is.na(custData_subset$income), meanIncome, custData_subset$income) -** fill missing value with mean.
- **summary(custData$income.fix) -** show there is no missing value

REPLACING OUTLIERS WITH MAX/MIN VALUES

- We believe income is not supposed to have negative values. We can replace the negative value(s) with 0.
- **custData_subset$income.fix<-ifelse(custData_subset$income.fix<0, 0, custData_subset$income.fix)**
- **summary(custData$income.fix)** -shows there is no negative value(s).

CONVERTING NUMERICAL DATA TO CATEGORICAL DATA

- **breaks <- c(0, 10000, 50000, 100000, 250000, 1000000)** - define income groups
- **custData_subset$income.groups <- cut(custData_subset$income.fix, breaks=breaks, include.lowest=T)** - cut the data into.
- Argument include.lowest=T is to make sure zero income data is included in the lowest group.

DATA TRANSFORMATION

- **medianincome <- read.table("median.income.csv", sep=',', header=T)**
- **custData_subset <- merge(custData_subset, medianincome, by.x="state.of.res", by.y="State")** - Merge
- **custData_subset$income.fix.norm <- custData_subset$income.fix / custData_subset$Median.Income** -Normalize the income by median income

EXERCISES

- Load Credit Risk dataset.
- Replace negative values in Age column with median age.
- Using IQR rule and empirical rule with -2.5σ and 2.5σ, determine the valid range of Credit.amount column Use only positive values when determining the valid range.
- Explain what to be done with the outliers in Credit.amount column.
- Replace negative values in Credit.amount column with median value.
- Derive a new attribute called Credit amount per duration attribute.