Code

MATH2349 Semester 2, 2018

Assignment 1

Vikas Virani - S3715555

Setup

Hide

```
library(readr) # Useful for importing data
library(foreign) # Useful for importing SPSS, SAS, STATA etc. data files
library(rvest) # Useful for scraping HTML data
library(knitr) # Useful for creating nice tables
library(magrittr) # Useful for using Pipe operator ( %>% )
```

Data Description

The dataset is of Presedential Elections for U.S. President from 1932 to 2016 in each State and the District of Columbia, Where the district of Columbia contibutes data from 1964 onwards & Hawaii and Alaska contibutes from 1960 onwards. Following is the source link of Dataset :-

- https://vincentarelbundock.github.io/Rdatasets/datasets.html (https://vincentarelbundock.github.io/Rdatasets/datasets.html)

Following are the variables in Dataset :-

1. Sequence of Number
2. state - character :- Name of all states of US
3. demVote - numeric :- Percentage of Vote for president won by Demographic Candidate
4. year - numeric :- Year in which election was held.
5. south :- Boolen value for whether or not a state is Confederacy state.

Read/Import Data

Hide

```
US_Elections <- read.csv("US Elections.csv")
head(US_Elections)
```

	X	state	demVote	year	south
	<int>	<fctr>	<dbl>	<int>	<lgl>
1	1	Alabama	84.76	1932	TRUE
2	2	Arizona	67.03	1932	FALSE
3	3	Arkansas	86.27	1932	TRUE

	X	state	demVote	year	south
	<int>	<fctr>	<dbl>	<int>	<lgl>
4	4	California	58.41	1932	FALSE
5	5	Colorado	54.81	1932	FALSE
6	6	Connecticut	47.40	1932	FALSE

6 rows

- Stored the downloaded dataset into current working directory.
- Base R library is used to use read.csv() function to read “Us Election” CSV file, so no need to use any library. As, the “StringAsfactors” is not defined as FALSE,it will by default take it TRUE and converts State Column of Dataset from Character to Factor.
- Using read.csv() function, stored the dataset into a varialble called “US_Elections”.
- Used head() function to display some data of “US_Elections” variable to check if data is properly loaded or not. As the first column does not have any name in csv file, it took a default X column name.

Inspect and Understand

- Used Class() and dim() functions to check if variable “US_Elections” is Data frame or not and dimensions of variable. It is a Data frame with 5 dimensions, i.e. 5 colums of the variable. And str() function to check the structure of data frame.
- Used Class() function on All 5 columns of Dataset to check the Data types. The data type of the variables are as follows :-

1. X - Integer
2. state - Factor
3. demVote - Numeric
4. year - Integer
5. south - Logical

- checked the levels of factor variable “state”, It correctly denotes 51 states of US and hence no need to change anything.
- The First column & column name demVote, year and south should be changed to meaningfull, relevant name. colnames() function is used to changes the name of these columns.

Hide

```
US_Elections %>% class()
```

```
[1] "data.frame"
```

Hide

```
US_Elections %>% dim()
```

```
[1] 1097    5
```

Hide

```
US_Elections %>% str()
```

```
'data.frame':   1097 obs. of  5 variables:
 $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ state  : Factor w/ 51 levels "Alabama","Alaska",..: 1 3 4 5 6 7 9 10 11 13 ...
 $ demVote: num  84.8 67 86.3 58.4 54.8 ...
 $ year   : int  1932 1932 1932 1932 1932 1932 1932 1932 1932 1932 ...
 $ south  : logi  TRUE FALSE TRUE FALSE FALSE FALSE ...
```

Hide

```
class(US_Elections$X)
```

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

Hide

```
class(US_Elections$state)
```

```
[1] "factor"
```

Hide

```
class(US_Elections$demVote)
```

```
[1] "numeric"
```

Hide

```
class(US_Elections$year)
```

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

Hide

```
class(US_Elections$south)
```

```
[1] "logical"
```

Hide

```
levels(US_Elections$state)
```

```
 [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"      
 [5] "California"     "Colorado"       "Connecticut"    "DC"            
 [9] "Delaware"       "Florida"        "Georgia"        "Hawaii"        
[13] "Idaho"          "Illinois"       "Indiana"        "Iowa"          
[17] "Kansas"         "Kentucky"       "Louisiana"      "Maine"         
[21] "Maryland"       "Massachusetts"  "Michigan"       "Minnesota"     
[25] "Mississippi"    "Missouri"       "Montana"        "Nebraska"      
[29] "Nevada"         "New Hampshire"  "New Jersey"     "New Mexico"    
[33] "New York"       "North Carolina" "North Dakota"   "Ohio"          
[37] "Oklahoma"       "Oregon"         "Pennsylvania"   "Rhode Island"  
[41] "South Carolina" "South Dakota"   "Tennessee"      "Texas"         
[45] "Utah"           "Vermont"        "Virginia"       "Washington"    
[49] "West Virginia"  "Wisconsin"      "Wyoming"       
```

Hide

```
colnames(US_Elections)[1]<-c("Number")
colnames(US_Elections)[3:5]<-c("Percentage Vote by Demographic Candidate","Election year","Confe
deracy State")
```

Subsetting I

- filtered first 10 rows of dataset with all columns by using “[1:10,]” subsetting and then piped the result in “as.matrix()” function to convert it into matrix named “Matrix_subset”.
- By using str() function, it is observed that Resulting “Matrix_subset” is of Character type. As Matrix can hold only one type of data, whole Matrix gets converted into Character data type to fit all values because character has higher precedence than number, i.e. character type can store the numeric values but not the vice versa.

Hide

```
Matrix_subset <- US_Elections[1:10,] %>% as.matrix()
str(Matrix_subset)
```

```
 chr [1:10, 1:5] " 1" " 2" " 3" " 4" " 5" " 6" " 7" " 8" " 9" "10" ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:10] "1" "2" "3" "4" ...
  ..$ : chr [1:5] "Number" "state" "Percentage Vote by Demographic Candidate" "Election year"
...
```

Subsetting II

- Again took a subset of all rows with 2nd and 5th column using component c(2,5), as first column is only a sequence number, to subset the data from “US_Elections” data frame.
- used save() function to save Subset_2 data frame as .RData file by providing the parameter “file”" to give filename to save() function. The resulting file will be stored in .RData format in current working directory with name “Subset_2.Rdata”.

Hide

```
Subset_2 <- US_Elections[,c(2,5)]
save(Subset_2,file="Subset_2.RData")
```

Create a new Data Frame

Created a data frame named “Data_Frame_1” using data.frame() function with one numeric variable (integer) named “numeric_vector” and one nominal (categorical) variable named “nominal_variable”.

- Nominal variable is a Factor and Ordered. when creating a data frame, “intensity” name is given that variable.
- I have used str() and levels() function to check the structure of variables and levels of Factor variable. “numeric_vector” is a number variable and “nominal_variable” is an ordinal variable with levels “Low”,“Medium”,“High” and “Extensive”.
- Created a numeric vector named “vector_1”” and used cbind() to add data frame - “Data_Frame_1” and new vector “vector_1” and stored resulting data frame into a variable called “New_Data_Frame”.
- Checked the attributes and the dimension of new data frame - “New_Data_Frame” using str(), attributes() and dim() functions. New data frame consists of 4 observations of 3 variables.
- As data frame can hold different data types for different columns, number vector remains number only when creating data frame or using cbind() and not gets converted to character unlike matrix. Following is the code and output :-

Hide

```
numeric_vector<-c(1,2,3,4) %>% as.integer()
vc_temp <- c("Low","Medium","High","Extensive")
nominal_variable <- ordered(x=vc_temp,c("Low","Medium","High","Extensive"))
Data_Frame_1 <- data.frame(number=numeric_vector,intensity=nominal_variable)
str(numeric_vector)
```

```
 int [1:4] 1 2 3 4
```

Hide

```
str(nominal_variable)
```

```
 Ord.factor w/ 4 levels "Low"<"Medium"<..: 1 2 3 4
```

Hide

```
nominal_variable %>% levels()
```

```
[1] "Low"       "Medium"    "High"      "Extensive"
```

Hide

```
vector_1 <- c(10,20,30,40)
New_Data_Frame <- cbind(Data_Frame_1,Value=vector_1)
New_Data_Frame %>% attributes()
```

```
$`names`
[1] "number"    "intensity" "Value"

$class
[1] "data.frame"

$row.names
[1] 1 2 3 4
```

Hide

```
str(New_Data_Frame)
```

```
'data.frame':	4 obs. of  3 variables:
 $ number   : int  1 2 3 4
 $ intensity: Ord.factor w/ 4 levels "Low"<"Medium"<..: 1 2 3 4
 $ Value    : num  10 20 30 40
```

Hide

```
dim(New_Data_Frame)
```

```
[1] 4 3
```