Exercise 3 - Data Manipulations using R

S.DEIVANAYKI (21BCS003)

30.01.2024

R Markdown

```
library(readr)
```

1. Use help and read in data from the file “Gcsemv.txt”. The data contain GCSE exam scores on a science subject. Two components of the exam were chosen as outcome variables : written paper and course work. There are 1905 students from 73 schools in England. Five fields are as follows. Missing values are coded as -1. 1. School ID 2. Student ID 3. Gender of student 0=boy 1=girl 4. Total score of written paper 5. Total score of coursework paper

```
df <- read_delim("C:/Users/DSL-A-B1/Downloads/RLAB/Gcsemv.txt",delim="\t")

head(df)

## # A tibble: 6 × 6
##   rownames school student gender written course
##      <dbl>  <dbl>   <dbl> <chr>    <dbl>  <dbl>
## 1        1  20920      16 M           23   NA
## 2        2  20920      25 F           NA   71.2
## 3        3  20920      27 F           39   76.8
```

a. Read in the data, give its summary and give appropriate names to the columns.

```
summary(df)

##     rownames         school         student        gender
##  Min.   :   1   Min.   :20920   Min.   :   1   Length:1905
##  1st Qu.: 477   1st Qu.:60501   1st Qu.:  64   Class :character
##  Median : 953   Median :68133   Median : 133   Mode  :character
##  Mean   : 953   Mean   :62128   Mean   :1037
##  3rd Qu.:1429   3rd Qu.:68411   3rd Qu.: 458


colnames(df) <-
c("S.No.","School_Id","Student_Id","Gender","Written_Score","Course_Score")
head(df)

## # A tibble: 6 × 6
##   S.No. School_Id Student_Id Gender Written_Score Course_Score
##   <dbl>     <dbl>      <dbl> <chr>          <dbl>        <dbl>
## 1     1     20920         16 M                 23         NA
## 2     2     20920         25 F                 NA         71.2
## 3     3     20920         27 F                 39         76.8
```

b. Handle the missing values.

```
fill_value1 <- mean(df$Written_Score,na.rm=TRUE)
fill_value2 <- mean(df$Course_Score,na.rm=TRUE)
df$Written_Score <-
ifelse(is.na(df$Written_Score),fill_value1,df$Written_Score)
df$Course_Score <- ifelse(is.na(df$Course_Score),fill_value2,df$Course_Score)
head(df)
```

```
## # A tibble: 6 × 6
##   S.No. School_Id Student_Id Gender Written_Score Course_Score
##   <dbl>     <dbl>      <dbl> <chr>          <dbl>        <dbl>
## 1     1     20920         16 M               23           73.4
## 2     2     20920         25 F               46.4         71.2
## 3     3     20920         27 F               39           76.8
```

Some of the variables read in as numeric are actually categorical variables. Convert them accordingly

```
df$Gender=as.factor(df$Gender)
head(df)
```

```
## # A tibble: 6 × 6
##   S.No. School_Id Student_Id Gender Written_Score Course_Score
##   <dbl>     <dbl>      <dbl> <fct>          <dbl>        <dbl>
## 1     1     20920         16 M               23           73.4
## 2     2     20920         25 F               46.4         71.2
## 3     3     20920         27 F               39           76.8
```

2. Write R command to Modify and one value in the above created, myiris.csv file by opening it in excel and compare the both (usingcomparedf, diffdf, all_equal, identical commands) and determine which value has been modified.

```
df1 <- read_csv("C:/Users/DSL-A-B1/Downloads/RLAB/myiris.csv")

head(df1)
```

```
## # A tibble: 6 × 6
##    ...1 Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>       <dbl>        <dbl>       <dbl> <chr>
## 1     1          5.1         3.5          1.4         0.2 setosa
## 2     2          4.9         3            1.4         0.2 setosa
## 3     3          4.7         3.2          1.3         0.2 setosa
```

```r
#install.packages("arsenal")
library(arsenal)
comparedf(df,df1)
```

```
## Compare Object
##
## Function Call:
## comparedf(x = df, y = df1)
##
## Shared: 0 non-by variables and 150 observations.
## Not shared: 12 variables and 1755 observations.
##
## Differences found in 0/0 variables compared.
## 0 variables compared have non-identical attributes.
```

```r
#install.packages("diffdf")
library(diffdf)
diffdf(df,df1)
```

```
## Warning in diffdf(df, df1):
## There are rows in BASE that are not in COMPARE !!
## A summary is given below.
##
## There are rows in BASE that are not in COMPARE !!
## First 10 of 1755 rows are shown in table below
##
##   ===============
##    ..ROWNUMBER..
##   ---------------
##         151
##         152
##         153
##         154
##
##   ===============
##       COLUMNS
##   ---------------
##        S.No.
##      School_Id
##     Student_Id
##       Gender
##    Written_Score
##    Course_Score
##   ---------------
##   ==============
##      COLUMNS
##   --------------
##        ...1
##    Sepal.Length
##    Sepal.Width
##    Petal.Length
##    Petal.Width
##      Species
##   --------------
```

```
all.equal(df,df1)
```

```
##  [1] "Names: 6 string mismatches"
##  [2] "Attributes: < Component \"row.names\": Numeric: lengths (1905, 150)
differ >"
##  [3] "Attributes: < Component \"spec\": Component \"cols\": Names: 6
string mismatches >"
##  [4] "Attributes: < Component \"spec\": Component \"cols\": Component 4:
mismatch >"
##  [7] "Component 1: Numeric: lengths (1905, 150) differ"
##  [8] "Component 2: Numeric: lengths (1905, 150) differ"
##  [9] "Component 3: Numeric: lengths (1905, 150) differ"
```

```
identical(df,df1)
```

```
## [1] FALSE
```

3. Write R script

1. From the given nation dataset

```
df2 <- read_csv("C:/Users/DSL-A-B1/Downloads/RLAB/nations1.csv")

df2$income <- ifelse(df2$income == "High income", 100000,ifelse(df2$income ==
"Low income", 25000,ifelse(df2$income == "Upper middle income", 75000,
40000)))
head(df2)
```

```
## # A tibble: 6 × 10
##   iso2c iso3c country  year life_expect population birth_rate
neonat_mortal_rate
##   <chr> <chr> <chr>   <dbl>       <dbl>      <dbl>      <dbl>
<dbl>
## 1 AD    AND   Andorra  1994          NA      62707       10.9
3.2
## 2 AD    AND   Andorra  1995          NA      63854       11
3
## 3 AD    AND   Andorra  2006          NA      83373       10.6
1.9
```

2. Filter only 2014 data and select columns for country, life expectancy, income group and region and save it as longevity.

```
library(tidyverse)
```

```
longevity <- df2 %>% filter(year==2014 & !is.na(life_expect)) %>%
select(country,life_expect,income,region)
head(longevity)
```

```
## # A tibble: 6 × 4
##   country              life_expect income region
##   <chr>                      <dbl>  <dbl> <chr>
## 1 United Arab Emirates        77.4 100000 Middle East & North Africa
## 2 Afghanistan                 60.4  25000 South Asia
## 3 Antigua and Barbuda         75.9  75000 Latin America & Caribbean
```

3. From it find the ten high income countries with the shortest life expectancy.

```
res <- longevity %>% arrange(desc(income))
head(res,10)
```

```
## # A tibble: 10 × 4
##    country              life_expect income region
##    <chr>                      <dbl>  <dbl> <chr>
##  1 United Arab Emirates        77.4 100000 Middle East & North Africa
##  2 Austria                     81.3 100000 Europe & Central Asia
##  3 Australia                   82.3 100000 East Asia & Pacific
```

4. Find countries in North America or Europe or Central Asia with a life expectancy in 2016 of 75-80.

```
ans4 <- df2 %>% filter(year==2014,(life_expect >= 75 & life_expect <=
80),region %in% c('North America','Europe & Central Asia'))
head(ans4)
```

```
## # A tibble: 6 × 10
##   iso2c iso3c country  year life_expect population birth_rate
neonat_mortal_rate
##   <chr> <chr> <chr>   <dbl>       <dbl>      <dbl>      <dbl>
<dbl>
## 1 AL    ALB   Albania  2014        77.8    2893654      13.4
6.5
## 2 BA    BIH   Bosnia…  2014        76.4    3817554       8.95
4.2
## 3 BG    BGR   Bulgar…  2014        75.4    7223938       9.4
5.9
## 4 CZ    CZE   Czech …  2014        78.3   10525347      10.4
1.9
```

5. Find the 20 countries with the longest life expectancies, plus the United states with its rank, if it lies outside the top 20.

```
c <- df2 %>% arrange(desc(life_expect)) %>% select(country) %>% unique()
dfc <- data.frame(Rank=1:nrow(c),Country=c)
```

```
result <- dfc %>% filter((Rank>=1 & Rank<=20) | country=='United States')
head(result)

##   Rank               country
## 1    1 Hong Kong SAR, China
## 2    2                 Japan
## 3    3            San Marino
```

6. Calculate the total GDP by income group and year and save it as a features. Sort the results in descending order of GDP.

```
features <- df2 %>% group_by(year) %>%
summarise(GDP=(sum(income,na.rm=TRUE)*100)/sum(population,na.rm=TRUE))
ans<- arrange(features,desc(GDP))
head(ans)

## # A tibble: 6 × 2
##    year   GDP
##   <dbl> <dbl>
## 1  1990 0.266
## 2  1991 0.261
## 3  1992 0.257
```

7. Summarize the data by year, finding the maximum and minimum country level life expectancies, and then Calculate the range of values.

```
ans1 <- df2 %>% group_by(year) %>%
summarise(MaxL=max(life_expect,na.rm=TRUE),MinL=min(life_expect,na.rm=TRUE),R
ange=(max(life_expect,na.rm = TRUE)-min(life_expect,na.rm = TRUE)))

head(ans1)

## # A tibble: 6 × 4
##    year  MaxL  MinL Range
##   <dbl> <dbl> <dbl> <dbl>
## 1  1990  78.8  33.5  45.4
## 2  1991  79.1  29.7  49.4
## 3  1992  79.2  27.5  51.7
```

8. Find total GDP in trillions of dollars, by region, over time.

```
res1 <- df2 %>% group_by(region,year) %>%
summarize(GDP=sum(income,na.rm=TRUE)/sum(population,na.rm=TRUE))

## `summarise()` has grouped output by 'region'. You can override using the
## `.groups` argument.

head(res1)
```

```
##   region               year     GDP
##   <chr>               <dbl>   <dbl>
## 1 East Asia & Pacific  1990 0.00127
## 2 East Asia & Pacific  1991 0.00125
## 3 East Asia & Pacific  1992 0.00124
```

9. Join nations to nation2 and total carbon dioxide , in gigatones , by region, over time.

```
df3 <- read.csv("C:/Users/DSL-A-B1/Downloads/nations2.csv")
colnames(df3)<-c('country_code','country','year','value')
head(df3)
```

```
##   country_code country year    value
## 1          ABW   Aruba 1960 11092.67
## 2          ABW   Aruba 1961 11576.72
```

```
newdf <- inner_join(df2,df3,by='country')
```

```
res3 <- newdf %>% group_by(year.x) %>% summarize(Total_CO2=sum(value))
head(res3)
```

```
## # A tibble: 6 × 2
##   year.x   Total_CO2
##    <dbl>       <dbl>
## 1   1990 1219261177.
## 2   1991 1219261177.
## 3   1992 1219261177.
```

4. Load the titanic dataset

```
titanic <- read.csv("C:/Users/DSL-A-B1/Downloads/Titanic.csv")
head(titanic)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
```

```
##                                                   Name    Sex Age SibSp
Parch
## 1                             Braund, Mr. Owen Harris   male  22     1
0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
0
## 3                              Heikkinen, Miss. Laina female  26     0
0
##             Ticket     Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
```

1. Find out number of samples missing age values

```
n <- sum(is.na(titanic$Age))
n

## [1] 177
```

2. Replace the missing fare value with median fare of the class.

```
titanic$Fare<-ifelse(is.na(titanic$Fare),median(titanic$Fare),titanic$Fare)
head(titanic)

##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
##                                                    Name    Sex Age SibSp
Parch
## 1                             Braund, Mr. Owen Harris   male  22     1
0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
0
## 3                              Heikkinen, Miss. Laina female  26     0
0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
```

3. Extract the surnames from the Passengers name.

```
ans3 <- titanic %>% transmute(Surname=str_extract(titanic$Name,"[^ ]+$"))
head(ans3)

##   Surname
## 1  Harris
## 2 Thayer)
## 3   Laina
```