Exercise 2 - Load and Explore the Data using R

S.DEIVANAYKI (21BCS003)

23.01.2024

R Markdown

#I. Write a R command to

#1. Install a package

```
#install.packages("ggplot2")
#library(ggplot2)
```

#2.Load a package

```
library(ggplot2)
```

#3.Unload a package

```
unloadNamespace("ggplot2")
```

#4.Remove an installed package from our system

```
#remove.packages("dplyr")
```

#5. Update a package

```
#update.packages("dplyr")

library(stringr)
library(tidyverse)

library(dplyr)
library(nycflights13)
```

#II. Load the inbuilt Iris dataset and explore the following

```
df<-data.frame(iris)
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

#1. Display the structure of the Iris Dataset

```r
str(df)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...
```

#2. Display the column names

```r
colnames(df)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

#3. Find the class of each column

```r
class(df$Sepal.Length)
```

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

```r
class(df$Sepal.Width)
```

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

```r
class(df$Petal.Length)
```

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

```r
class(df$Petal.Width)
```

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

```r
class(df$Species)
```

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

#4. Display the first 10 rows of the dataset

```r
head(df,n=5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

#5. Display last 10 data of the feature Sepal.Length

```r
tail(df$Sepal.Length,5)
```

```
## [1] 6.7 6.3 6.5 6.2 5.9
```

#6. Produce a summary of the Petal.Length

```
summary(df$Petal.Length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.600   4.350   3.758   5.100   6.900
```

#7. Find out number of samples in each class of iris dataset.

```
df %>% group_by(Species) %>% summarize(Count=n())
```

```
## # A tibble: 3 × 2
##   Species    Count
##   <fct>      <int>
## 1 setosa        50
## 2 versicolor    50
## 3 virginica     50
```

#8. Write the iris data as a "myiris.csv"

```
write.csv(df,file="myiris.csv")
```

#III. Load the Titanic dataset

```
t1 <- read.csv("C:/Users/DSL-A-B1/Downloads/Titanic.csv")
head(t1)
```

```
##   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
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. Examine the dataset using glimpse

```
glimpse(t1)
```

```
## Rows: 891
## Columns: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17,…
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1,
0, 1…
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2,
3, 3…
```

#2. Check if duplicate entries exists and drop passenger id.

```
dup<-t1[!duplicated(t1$PassengerId),]
head(dup)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      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
```

#3. Create SurvivedFac(as categorical outcome) and SurvivedNum (as range from 0 to 1) and drop Survived.

```
df<-t1
df$SurvivedFac <- factor(df$Survived,levels=c(1,0),labels=c("Yes","No"))
df$SurvivedNum <- as.numeric(df$Survived)
head(select(df,-Survived))
```

```
##   PassengerId Pclass                                                 Name
Sex
## 1           1      3                              Braund, Mr. Owen Harris
male
## 2           2      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer)
female
## 3           3      3                               Heikkinen, Miss. Laina
male
##   Age SibSp Parch           Ticket    Fare Cabin Embarked SurvivedFac
## 1  22     1     0        A/5 21171  7.2500              S          No
## 2  38     1     0         PC 17599 71.2833   C85        C         Ye
##   SurvivedNum
## 1           0
## 2           1
```

```
## 3           1
```

#4. Create PClassFac (as categorical) and PClassNum (as ordinal) and drop PClass.

```
df$PClassFac <- factor(df$Pclass)
df$PClassNum<-as.numeric(df$Pclass)
df<-select(df,-df$PClass)
head(df)
```

```
##   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
PClassFac
## 1        A/5 21171  7.2500              S          No           0
3
## 2         PC 17599 71.2833   C85        C         Yes           1
1
##   PClassNum
## 1         3
## 2         1
## 3         3
```

#5. Show title count by gender and combine title with low count as rare.

```
library(stringr)
t1$Title<-
ifelse(str_detect(t1$Name,"Miss"),"Miss",(ifelse(str_detect(t1$Name,"Mr"),"Mr
","Rare")))
head(t1)
```

```
##   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 Title
## 1        A/5 21171  7.2500              S    Mr
## 2         PC 17599 71.2833   C85        C    Mr
## 3 STON/O2. 3101282  7.9250              S  Miss
```

#6. Create a family size variable including the passenger themselves.

```
t1$FamilySize <- t1$SibSp+1
head(t1)
```

```
##   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 Title FamilySize
## 1        A/5 21171  7.2500              S    Mr          2
## 2         PC 17599 71.2833   C85        C    Mr          2
## 3 STON/O2. 3101282  7.9250              S  Miss          1
```

#7. Create new age dependent variables: Chlid and Mother. A Child will simply be someone under 18 years of age and a mother is passenger who is 1) female 2) over 18 3) more than 0 children 4) doesn't miss

```
t1$Child <-ifelse(t1$Age<18,"Yes","No")
t1$Mother <- ifelse(t1$Sex=="Female" & t1$Age>18 & t1$SibSp>0 &
str_detect(t1$Name,"Miss"),"Yes","No")
head(t1)
```

```
##   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 Title FamilySize Child Mother
## 1        A/5 21171  7.2500              S    Mr          2    No     No
## 2         PC 17599 71.2833   C85        C    Mr          2    No     No
## 3 STON/O2. 3101282  7.9250              S  Miss          1    No     No
```

#IV. Do the following in R

#1. Using Flights data

```
t1 <- flights %>% summarise(Travel_long = max(arr_time-dep_time,na.rm=TRUE))
t1 <- as.numeric(t1)
head(filter(flights,(flights$arr_time-flights$dep_time)==t1))
```

```
## # A tibble: 1 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     3    23      959            920        39     2129
1240

t2 <- flights %>% summarise(Travel_short = min(arr_time-dep_time,na.rm=TRUE))
t2 <- as.numeric(t2)
head(filter(flights,(flights$arr_time-flights$dep_time)==t2))
```

```
## # A tibble: 1 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     7    17     2400           2142       138       54
2259
```

#2. Using Flights dataframe calculate gain = arr_delay - dep_delay and gain_per_hour.

```
df <- mutate(flights,gain = arr_delay - dep_delay,gain_per_hour =
(gain*60)/air_time)
head(df)
```

```
## # A tibble: 6 × 21
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     1      517            515         2      830
819
## 2  2013     1     1      533            529         4      850
830
```

```
## 3  2013     1     1      542            540         2      923
850
## 4  2013     1     1      544            545        -1     1004
1022
```

#3. Find the total number of miles a plane flew

```
head(select(flights,flight,distance))
```

```
## # A tibble: 6 × 2
##   flight distance
##    <int>    <dbl>
## 1   1545     1400
## 2   1714     1416
## 3   1141     1089
```

#4. How many flights left before 5am?

```
f <- flights %>% filter(dep_time < 500)
head(f)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     2       42           2359        43      518
442
## 2  2013     1     2      126           2250       156      233
2359
## 3  2013     1     2      458            500        -2      703
650
```

#5. Group flights by destination.

```
f1 <- flights %>% group_by(dest)
head(f1)
```

```
## # A tibble: 6 × 19
## # Groups:   dest [5]
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     1      517            515         2      830
819
## 2  2013     1     1      533            529         4      850
830
## 3  2013     1     1      542            540         2      923
```

```
850
## 4  2013     1     1      544            545        -1     1004
1022
```

#6. Summarize to compute distance, average delay and number of flights

```
paste("Compute distance")
```

```
## [1] "Compute distance"
```

```
head(flights %>% select(flight,distance))
```

```
## # A tibble: 6 × 2
##   flight distance
##    <int>    <dbl>
## 1   1545     1400
## 2   1714     1416
## 3   1141     1089
```

```
head(summarise(flights,Avg_delay=mean(arr_delay+dep_delay,na.rm=TRUE)))
```

```
## # A tibble: 1 × 1
##   Avg_delay
##       <dbl>
## 1      19.5
```

```
head(summarise(flights,No_Of_Flight=n()))
```

```
## # A tibble: 1 × 1
##   No_Of_Flight
##          <int>
## 1       336776
```

#7. Compare air_time with arr_time - dep_time.

```
Result<-data.frame(Result=ifelse(flights$air_time>(flights$arr_time-
flights$dep_time),"Greater","Lesser"))
head(mutate(flights,Result))
```

```
## # A tibble: 6 × 20
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     1      517            515         2      830
819
## 2  2013     1     1      533            529         4      850
830
## 3  2013     1     1      542            540         2      923
```

```
850
```

#8. Find average delay per month.

```
a <- flights %>% group_by(month) %>% summarize(Avg_delay =
mean(arr_delay+dep_delay,na.rm=TRUE))
head(a)
```

```
## # A tibble: 6 × 2
##   month Avg_delay
##   <int>     <dbl>
## 1     1      16.1
## 2     2      16.4
## 3     3      19.0
```

#9. When do the first and last flights leave each day?

```
df <- flights %>% group_by(day) %>% summarize(Last=max(arr_time,na.rm=TRUE))
t2 <- filter(flights,flights$day==df$day,flights$arr_time==df$Last)
```

```
## Warning: There were 2 warnings in `filter()`.
## The first warning was:
## ℹ In argument: `flights$day == df$day`.
## Caused by warning in `flights$day == df$day`:
## ! longer object length is not a multiple of shorter object length
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
head(t2)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013    10    24     2131           2134        -3     2400
2
## 2  2013    10    29     2124           2130        -6     2400
18
## 3  2013    12    30     2023           2021         2     2400
2357
<dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
df1 <- flights %>% group_by(day) %>%
summarize(First=min(arr_time,na.rm=TRUE))
t3 <- filter(flights,flights$day==df1$day,flights$arr_time==df1$First)
```

.

```
head(t3)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1    16     2018           2025        -7        1
2329
## 2  2013     1    24     2309           2250        19        1
2354
## 3  2013     3    16     2122           2009        73        1
2240
```

#10. Which destination have the most carriers?

```
f1 <- flights %>% mutate(dest) %>% summarize(Most_carrier=max(carrier))
head(f1)

## # A tibble: 1 × 1
##   Most_carrier
##   <chr>
## 1 YV
```

#11. Find all flights departed between midnight and 6am

```
f2 <- filter(flights,dep_time>=0 & dep_time<=600)
head(f2)

## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     1      517            515         2      830
819
## 2  2013     1     1      533            529         4      850
830
## 3  2013     1     1      542            540         2      923
850
```

#12. Find the flights that left earliest

```
df1 <- flights %>% group_by(day) %>%
summarize(First=min(arr_time,na.rm=TRUE))
f3 <- filter(flights,flights$day==df1$day,flights$arr_time==df1$First)

## Warning: There were 2 warnings in `filter()`.
## The first warning was:
## ℹ In argument: `flights$day == df1$day`.
## Caused by warning in `flights$day == df1$day`:
```

```
## ! longer object length is not a multiple of shorter object length
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
head(f3)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1    16     2018           2025        -7        1
2329
## 2  2013     1    24     2309           2250        19        1
2354
## 3  2013     3    16     2122           2009        73        1
2240
## 4  2013     3    29     2055           2100        -5        1
32
## 5  2013     7     3     2142           2100        42        1
2336
## 6  2013     8     3     2053           2025        28        1
2321
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance
<dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

#13. What proportion of flights are delayed by more than an hour.

```
f4 <- filter(flights,flights$arr_delay>60,na.rm=TRUE)
head(f4)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
<int>
## 1  2013     1     1      811            630       101     1047
830
## 2  2013     1     1      848           1835       853     1001
1950
```

#14. Find all groups where the count of flights is greater than 365.

```
f5 <- flights %>% group_by(flight) %>% summarise(count=n())
f6 <- filter(f5,f5$count>365)
head(f6)
```

```
## # A tibble: 6 × 2
##   flight count
```

```
##    <int> <int>
## 1      1   701
## 2      3   631
## 3      4   393
```

#15. Combine year,month,day,hour,minute fields to a single field departure

```
f7 <- unite(flights,"Departure",year,month,day,hour,minute,sep='-')
head(f7)
```

```
## # A tibble: 6 × 15
##   Departure  dep_time sched_dep_time dep_delay arr_time sched_arr_time
arr_delay
##   <chr>         <int>          <int>     <dbl>    <int>          <int>
<dbl>
## 1 2013-1-1-…      517            515         2      830            819
11
## 2 2013-1-1-…      533            529         4      850            830
20
```

#Additional

#1. Create an employee dataset with columns as age,edu,marital,income,ls,wkabint

```
data <-
data.frame(gender=sample(c(0,1),50,replace=TRUE),age=sample(20:45,50,replace=
TRUE),edu=sample(factor(c('UG','PG')),50,replace =
TRUE),marital=sample(factor(c("UnMarried","Married")),50,replace=TRUE),income
=sample(c(100000,50000,125000,25000,75000),50,replace=TRUE),ls=sample(factor(
c(1,2,3,4,5)),50,replace=TRUE),wkabint=sample(factor(c("Yes","No")),50,replac
e=TRUE))
head(data)
```

```
##   gender age edu   marital income ls wkabint
## 1      1  22  PG   Married  50000  4      No
## 2      0  41  UG   Married 125000  5      No
## 3      0  32  PG   Married  25000  2     Yes
```

#2. Update tibble with new column income_factor (income/age)

```
df1 <- mutate(data,Income_Factor=data$income/data$age)
head(df1)
```

```
##   gender age edu   marital income ls wkabint Income_Factor
## 1      1  22  PG   Married  50000  4      No      2272.727
## 2      0  41  UG   Married 125000  5      No      3048.780
## 3      0  32  PG   Married  25000  2     Yes       781.250
```

#3. Maximum Age of UG graduate in each level of ls.

```
data %>% group_by(ls) %>% filter(edu=='UG') %>% summarise(Max_Age_UG =
max(age))
```

```
## # A tibble: 5 × 2
##   ls    Max_Age_UG
##   <fct>      <int>
## 1 1             43
## 2 2             44
## 3 3             38
```