Stat 291 - Recitation 4

Orçun Oltulu

19 / 11 / 2021

Data Frames

Exercise 1:

Part A.

Create following data frame;

```
Name <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")
Age <- c(25, 31, 23, 52, 76, 49, 26)
Height <- c(177, 163, 190, 179, 163, 183, 164)
Weight <- c(57, 69, 83, 75, 70, 83, 53)
Gender <- as.factor(c("F", "F", "M", "M", "F", "M", "F"))
```

```
df <- data.frame(Name, Age, Height, Weight, Gender)
df
```

```
##       Name Age Height Weight Gender
## 1     Alex  25    177     57      F
## 2    Lilly  31    163     69      F
## 3     Mark  23    190     83      M
## 4   Oliver  52    179     75      M
## 5   Martha  76    163     70      F
## 6    Lucas  49    183     83      M
## 7 Caroline  26    164     53      F
```

Part B.

Check the dimensions of the data frame.

```
dim(df)
```

```
## [1] 7 5
```

Part C.

Check the class of each variable in the data frame.

```
class(df$Name);class(df$Age);class(df$Height);class(df$Weight);class(df$Gender)
```

```
## [1] "character"
```

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

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

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

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

```
# Alternatively,
str(df)
```

```
## 'data.frame':    7 obs. of  5 variables:
##  $ Name  : chr  "Alex" "Lilly" "Mark" "Oliver" ...
##  $ Age   : num  25 31 23 52 76 49 26
##  $ Height: num  177 163 190 179 163 183 164
##  $ Weight: num  57 69 83 75 70 83 53
##  $ Gender: Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1
```

Part D.

Remove 'Height' column from the data frame and print the new version.

```
df2 <- subset(df, select = -Height)
df2
```

```
##       Name Age Weight Gender
## 1     Alex  25     57      F
## 2    Lilly  31     69      F
## 3     Mark  23     83      M
## 4   Oliver  52     75      M
## 5   Martha  76     70      F
## 6    Lucas  49     83      M
## 7 Caroline  26     53      F
```

Part E.

Add a new column to the right side of your updated data frame named 'Working';

```
Working <- factor(c("Yes", "Yes", "No", "Yes", "No", "No", "Yes"))
```

```
df2 <- cbind(df2, Working)
df2
```

```
##       Name Age Weight Gender Working
## 1     Alex  25     57      F     Yes
## 2    Lilly  31     69      F     Yes
## 3     Mark  23     83      M      No
## 4   Oliver  52     75      M     Yes
## 5   Martha  76     70      F      No
## 6    Lucas  49     83      M      No
## 7 Caroline  26     53      F     Yes
```

Part F.

Calculate the mean 'Weight' in the data frame.

```
mean(df2$Weight)
```

```
## [1] 70
```

Part G.

What is the proportion of Working group?

```
(table(df2$Working)/nrow(df2))[2]
```

```
##       Yes
## 0.5714286
```

```
# Alternatively,
# nrow(df2[df2$Working == "Yes",]) / nrow(df2)
```

Part H.

Create a subset from the data frame consisting only Females, and has only 2 columns, Name and Age.

```
new_subset <- subset(df2, select = c(Name, Age), subset = Gender == "F")
new_subset
```

```
##       Name Age
## 1     Alex  25
## 2    Lilly  31
## 5   Martha  76
## 7 Caroline  26
```

Part I.

Print the names of people who are younger than 30.

```r
df2[df2$Age < 30,"Name"]
```

```
## [1] "Alex"     "Mark"     "Caroline"
```

```r
# Alternatively,
# subset(df2, subset=(Age > 30))[,"Name"]
# Alternatively,
# subset(df2, select=c(Name), subset=(Age > 30))
```

Exercise 2:

Part A.

Assume that there are 8 students taking the same course. There are also 2 different sections in this course. Use the same names in exercise 1 and add your name at the end. Now create 2 data frames as following;

```r
Name <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")
Name <- c(Name,"Orcun")
sections <- rep(1:2,each = 4)

section1 <- data.frame(Name = Name[1:4],Section = sections[1:4]); section1
```

```
##     Name Section
## 1   Alex       1
## 2  Lilly       1
## 3   Mark       1
## 4 Oliver       1
```

```r
section2 <- data.frame(Name = Name[5:8],Section = sections[5:8]); section2
```

```
##       Name Section
## 1   Martha       2
## 2    Lucas       2
## 3 Caroline       2
## 4    Orcun       2
```

```r
section1
```

```
##     Name Section
## 1   Alex       1
## 2  Lilly       1
## 3   Mark       1
## 4 Oliver       1
```

```r
section2
```

```
##       Name Section
```

```
## 1   Martha       2
## 2    Lucas       2
## 3 Caroline       2
## 4    Orcun       2
```

Part B.

Combine the rows of the two data frames in Part A and name it 'course'.

```
course <- rbind(section1, section2)
course
```

```
##       Name Section
## 1     Alex       1
## 2    Lilly       1
## 3     Mark       1
## 4   Oliver       1
## 5   Martha       2
## 6    Lucas       2
## 7 Caroline       2
## 8    Orcun       2
```

Part C.

Now assume there was a quiz, and the ones in the first section took the 'Quiz1' and the others took 'Quiz2'. Create another data frame and name it 'quiz'.

```
quiz <- data.frame(Name = Name,
                   Quiz = paste("Quiz",rep(1:2,each = 4),sep=""),
                   Grades = c(43, 17, 73, 23, 67, 97, 69, 100))
quiz
```

```
##       Name  Quiz Grades
## 1     Alex Quiz1     43
## 2    Lilly Quiz1     17
## 3     Mark Quiz1     73
## 4   Oliver Quiz1     23
## 5   Martha Quiz2     67
## 6    Lucas Quiz2     97
## 7 Caroline Quiz2     69
## 8    Orcun Quiz2    100
```

Part C.

Use the merge() function to merge the two data frames by "Name" into a new data frame, "course2".

```
course2 <- merge(course, quiz, by = "Name")
# sort = FALSE option
course2
```

```
##       Name Section  Quiz Grades
## 1     Alex       1 Quiz1     43
## 2 Caroline       2 Quiz2     69
## 3    Lilly       1 Quiz1     17
## 4    Lucas       2 Quiz2     97
## 5     Mark       1 Quiz1     73
## 6   Martha       2 Quiz2     67
## 7   Oliver       1 Quiz1     23
## 8    Orcun       2 Quiz2    100
```

Part D.

Print the Names of students whose quiz score is more than 70.

```
subset(course2, select = Name, subset = Grades > 70)
```

```
##    Name
## 4 Lucas
## 5  Mark
## 8 Orcun
```

Exercise 3:

Install and load 'ISLR' and 'dplyr' packages, then load 'iris' dataset by using following code;

```
#install.packages("ISLR")
#install.packages("dplyr")
library(ISLR) # iris data set
library(dplyr) # data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data("iris")
```

Part A.

Check the first 6 rows of iris data set.

```
head(iris)
```

```
##   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
```

Part B.

Check dimensions and types of variables of iris data set.

```
str(iris)
```

```
## '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 ..
```

Part C.

Select Sepal width and Species variables from iris data set and create a new data frame.

```
new_iris1 <- select(iris, c("Sepal.Width", "Species"))
head(new_iris1,10)
```

```
##    Sepal.Width Species
## 1          3.5  setosa
## 2          3.0  setosa
## 3          3.2  setosa
## 4          3.1  setosa
## 5          3.6  setosa
## 6          3.9  setosa
## 7          3.4  setosa
## 8          3.4  setosa
## 9          2.9  setosa
## 10         3.1  setosa
```

Part D.

Now, create another data frame where you have only 'virginica' species.

```
new_iris2 <- filter(iris, Species == "virginica")
head(new_iris2,10)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1           6.3         3.3          6.0         2.5 virginica
## 2           5.8         2.7          5.1         1.9 virginica
## 3           7.1         3.0          5.9         2.1 virginica
## 4           6.3         2.9          5.6         1.8 virginica
## 5           6.5         3.0          5.8         2.2 virginica
## 6           7.6         3.0          6.6         2.1 virginica
## 7           4.9         2.5          4.5         1.7 virginica
## 8           7.3         2.9          6.3         1.8 virginica
## 9           6.7         2.5          5.8         1.8 virginica
## 10          7.2         3.6          6.1         2.5 virginica
```

Part E.

Create a new data frame where you have;

- Sepal.Length > 5.5,
- Sepal.Width < 2.5.

```
new_iris3 <- filter(iris,
                    Sepal.Length > 5.5,
                    Sepal.Width < 2.5)

head(new_iris3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1          6.0         2.2          4.0         1.0 versicolor
## 2          6.2         2.2          4.5         1.5 versicolor
## 3          6.3         2.3          4.4         1.3 versicolor
## 4          6.0         2.2          5.0         1.5  virginica
```

Part F.

Create new data frame where you have Sepal.Length is in descending order.

```
new_iris4 <- arrange(iris, desc(Sepal.Length))
head(new_iris4)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1          7.9         3.8          6.4         2.0 virginica
```

```
## 2          7.7         3.8          6.7         2.2 virginica
## 3          7.7         2.6          6.9         2.3 virginica
## 4          7.7         2.8          6.7         2.0 virginica
## 5          7.7         3.0          6.1         2.3 virginica
## 6          7.6         3.0          6.6         2.1 virginica
```

Part G.

Subset only the 'numeric' variable in iris data set.

```
new_iris5 <- select_if(iris, is.numeric)
head(new_iris5, 10)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1           5.1         3.5          1.4         0.2
## 2           4.9         3.0          1.4         0.2
## 3           4.7         3.2          1.3         0.2
## 4           4.6         3.1          1.5         0.2
## 5           5.0         3.6          1.4         0.2
## 6           5.4         3.9          1.7         0.4
## 7           4.6         3.4          1.4         0.3
## 8           5.0         3.4          1.5         0.2
## 9           4.4         2.9          1.4         0.2
## 10          4.9         3.1          1.5         0.1
```

Part H.

Find mean of Sepal.Length and Sepal.Width for each 'Species' separately.

```
summarize(group_by(iris, Species),
          S.Length_mean = mean(Sepal.Length),
          S.Width_mean = mean(Sepal.Width))
```

```
## # A tibble: 3 x 3
##   Species    S.Length_mean S.Width_mean
##   <fct>              <dbl>        <dbl>
## 1 setosa              5.01         3.43
## 2 versicolor          5.94         2.77
## 3 virginica           6.59         2.97
```

Exercise 4:

Load 'Credit' data set from ISLR package. Read the document for the data set; '?Credit'.

Part A.

Check the first 10 observations of Credit data set.

```
head(Credit, 10)
```

```
##    ID  Income Limit Rating Cards Age Education Gender Student Married
## 1   1  14.891  3606    283     2  34        11   Male      No     Yes
## 2   2 106.025  6645    483     3  82        15 Female     Yes     Yes
## 3   3 104.593  7075    514     4  71        11   Male      No      No
## 4   4 148.924  9504    681     3  36        11 Female      No      No
## 5   5  55.882  4897    357     2  68        16   Male      No     Yes
## 6   6  80.180  8047    569     4  77        10   Male      No      No
## 7   7  20.996  3388    259     2  37        12 Female      No      No
## 8   8  71.408  7114    512     2  87         9   Male      No      No
## 9   9  15.125  3300    266     5  66        13 Female      No      No
## 10 10  71.061  6819    491     3  41        19 Female     Yes     Yes
##           Ethnicity Balance
## 1         Caucasian     333
## 2             Asian     903
## 3             Asian     580
## 4             Asian     964
## 5         Caucasian     331
## 6         Caucasian    1151
## 7  African American     203
## 8             Asian     872
## 9         Caucasian     279
## 10 African American    1350
```

Part B.

Create a subset, new_credit1, for Asian married females.

```
new_credit1 <- filter(Credit,
                      Gender == "Female",
                      Married == "Yes",
                      Ethnicity == "Asian")

head(new_credit1,10)
```

```
##    ID  Income Limit Rating Cards Age Education Gender Student Married Ethnicity
## 1   2 106.025  6645    483     3  82        15 Female     Yes     Yes     Asian
## 2  13  80.616  5308    394     1  57         7 Female      No     Yes     Asian
## 3  18  36.496  4378    339     3  69        15 Female      No     Yes     Asian
## 4  19  49.570  6384    448     1  28         9 Female      No     Yes     Asian
## 5  35  20.150  2646    199     2  25        14 Female      No     Yes     Asian
## 6  43  44.158  4763    351     2  66        13 Female      No     Yes     Asian
## 7  44  36.929  6257    445     1  24        14 Female      No     Yes     Asian
## 8  47  19.531  5043    376     2  64        16 Female     Yes     Yes     Asian
```

```
## 9  55  15.333  1499    138     2  47         9 Female      No     Yes     Asian
## 10 56  32.916  1786    154     2  60         8 Female      No     Yes     Asian
##    Balance
## 1      903
## 2      204
## 3      368
## 4      891
## 5        0
## 6      385
## 7      976
## 8     1241
## 9        0
## 10       0
```

Part C.

From new_credit1, select only numeric values and create new data frame, new_credit2.

```
new_credit2 <- select_if(new_credit1, is.numeric)
head(new_credit2,10)
```

```
##    ID  Income Limit Rating Cards Age Education Balance
## 1   2 106.025  6645    483     3  82        15     903
## 2  13  80.616  5308    394     1  57         7     204
## 3  18  36.496  4378    339     3  69        15     368
## 4  19  49.570  6384    448     1  28         9     891
## 5  35  20.150  2646    199     2  25        14       0
## 6  43  44.158  4763    351     2  66        13     385
## 7  44  36.929  6257    445     1  24        14     976
## 8  47  19.531  5043    376     2  64        16    1241
## 9  55  15.333  1499    138     2  47         9       0
## 10 56  32.916  1786    154     2  60         8       0
```

Part D.

Find the mean of each numeric value in new_credit2.

```
summarise_all(new_credit2, mean)
```

```
##         ID   Income    Limit Rating    Cards     Age Education  Balance
## 1 171.9524 49.54781 4842.548    361 2.595238 51.7619  13.61905 533.9762
```

Part F.

Now, find minimum and maximum Income for Asian males and females separately,

- Filter Asian people,

- Select only Gender and Income variables,
- Group them by Gender,
- Summarize using min and max functions.

```
new_credit3 <- filter(Credit, Ethnicity == "Asian")
new_credit3 <- select(new_credit3, Gender, Income)
new_credit3 <- group_by(new_credit3, Gender)

summarise(new_credit3,
          Minimum_Income = min(Income),
          Maximum_Income = max(Income))
```

```
## # A tibble: 2 x 3
##   Gender   Minimum_Income Maximum_Income
##   <fct>             <dbl>          <dbl>
## 1 " Male"            10.4           129.
## 2 "Female"           10.4           180.
```