Stat 291 - Recitation 5

Orçun Oltulu

26 / 11 / 2021

Last Week:

Recall the last exercise from Recitation 4.

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

```
library(ISLR)
library(dplyr)
```

```
##
## 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
```

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.
```

Reading data files into R:

Exercise 1:

Read the file 'Table0.txt';

```
df1 <- read.table("Table0.txt")
df1
```

```
##         V1 V2  V3 V4 V5
## 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
```

(a) Assign names to the columns to Name, Age, Height, Weight and Sex.

```
colnames(df1) <- c('Name', 'Age', 'Height', 'Weight','Sex');
df1
```

```
##       Name Age Height Weight Sex
## 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
```

(b) Change the row names so that they are the same as Name, and remove the variable Name.

```
row.names(df1) <- df1$Name
df1$Name <- NULL
df1
```

```
##          Age Height Weight Sex
## Alex      25    177     57   F
## Lilly     31    163     69   F
## Mark      23    190     83   M
## Oliver    52    179     75   M
## Martha    76    163     70   F
## Lucas     49    183     83   M
## Caroline  26    164     53   F
```

Exercise 2:

Read the file 'Table1.txt';

```
df2 <- read.table("Table1.txt",header=T)
df2
```

```
##       Name Age Height Weight Sex
## 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
```

(a) How many rows and columns does it have?

```
ncol(df2)
```

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

```
nrow(df2)
```

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

(b) Reread the file and make the variable Name be the row names. Make sure you read the variable as characters and not as factors.

```
df2.b <- read.table("Table1.txt", header=T,
                    row.names = "Name",
                    stringsAsFactors = FALSE)
df2.b
```

```
##          Age Height Weight Sex
## Alex      25    177     57   F
## Lilly     31    163     69   F
## Mark      23    190     83   M
## Oliver    52    179     75   M
## Martha    76    163     70   F
```

```
## Lucas     49    183     83   M
## Caroline  26    164     53   F
```

```
lapply(df2.b, class)
```

```
## $Age
## [1] "integer"
##
## $Height
## [1] "integer"
##
## $Weight
## [1] "integer"
##
## $Sex
## [1] "character"
```

Exercise 3:

Read the file 'Table2.txt';

```
df3 <- read.table('Table2.txt',
                  header = T,
                  skip = 1)
```

(a) What is the problem with the first and last columns ?

```
df3
```

```
##          Name Age Height Weight Sex
## 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/
```

(b) How can you fix that problem ?

```
df3.b <- read.table('Table2.txt',
                  header = T,
                  skip = 1, quote ="/")
df3.b
```

```
##       Name Age Height Weight Sex
## 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
```

Exercise 4:

Read the file 'Table3.txt';

```
df4 <- read.table('Table3.txt',header = T,skip = 1)
df4
```

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

(a) How many missing value does this data set have?

```
sum(is.na(df4))
```

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

(b) Assign NA to each 'weird' value.

```
df4[3,2] <- df4[5,3] <- df4[6,4] <- NA
df4
```

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

(c) Reread the data set but this time make sure you only have 'NA' values rather than {'*','**', "–"}

```
df4.c <- read.table('Table3.txt', header = T, skip = 1,
                    na.strings = c("NA", "*", "**", "--"))
df4.c
```

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

Exercise 5:

Read the file 'Table4.txt';

```
df5 <- read.table('Table4.txt',header = T)
df5
```

```
##       Name Age Height Weight Sex
## 1     Alex  25   1,77     57   F
## 2    Lilly  31   <NA>     69   F
## 3     Mark  --   1,90     83   M
## 4   Oliver  52   1,79     75   M
## 5   Martha  76      *     70   F
## 6    Lucas  49   1,83     **   M
## 7 Caroline  26   1,64     53   F
```

Watch out for the missing values and the decimal separator.

```
df5.a <- read.table('Table4.txt',header = T,
                    na.strings = c("NA", "*", "**","--"),
                    dec = ",")
df5.a
```

```
##       Name Age Height Weight Sex
## 1     Alex  25   1.77     57   F
## 2    Lilly  31     NA     69   F
## 3     Mark  NA   1.90     83   M
## 4   Oliver  52   1.79     75   M
## 5   Martha  76     NA     70   F
## 6    Lucas  49   1.83     NA   M
## 7 Caroline  26   1.64     53   F
```

Exercise 6:

Read the file 'Table5.txt';

```
df6 <- read.table('Table5.txt', header = T)
df6
```

```
##   Name.Age.Height.Weight.Sex
## 1          Alex;25;1,77;57;F
## 2           Lilly;31;NA;69;F
## 3          Mark;--;1,90;83;M
## 4        Oliver;52;1,79;75;M
## 5           Martha;76;;70;F
## 6         Lucas;49;1,83;**;M
## 7      Caroline;26;1,64;53;F
```

Watch out for the missing values and the decimal separator and the separator.

```
df6.a <- read.table('Table5.txt', header = T,
                    na.strings = c(NA, "**", "--"),
                    dec = ",", sep = ";")
df6.a
```

```
##       Name Age Height Weight Sex
## 1     Alex  25   1.77     57   F
## 2    Lilly  31     NA     69   F
## 3     Mark  NA   1.90     83   M
## 4   Oliver  52   1.79     75   M
## 5   Martha  76     NA     70   F
## 6    Lucas  49   1.83     NA   M
## 7 Caroline  26   1.64     53   F
```

Exercise 7:

Read the file 'Table6.txt';

Check out the file first. Notice that the information is repeated, we only want the first non-repeated ones. Make sure to create only characters not factors this time around. Lastly, we don't want the comments.

```
df7.a<- read.table("Table6.txt", skip = 1,header = TRUE,
                   row.names = "Name",nrow = 7,
                   comment.char = "@",
                   stringsAsFactors = FALSE)
df7.a
```

```
##          Age Height Weight Sex
## Alex      25    177     57   F
## Lilly     31    163     69   F
## Mark      23    190     83   M
## Oliver    52    179     75   M
## Martha    76    163     70   F
## Lucas     49    183     83   M
## Caroline  26    164     53   F
```

Exercise 8:

Read the file 'states1.csv';

```
df8 <- read.csv("states1.csv")
head(df8,10)
```

```
##              X Population Income Illiteracy Life.Exp Murder HS.Grad Frost
## 1      Alabama       3615   3624        2.1    69.05   15.1    41.3    20
## 2       Alaska        365   6315        1.5    69.31   11.3    66.7   152
## 3      Arizona       2212   4530        1.8    70.55    7.8    58.1    15
## 4     Arkansas       2110   3378        1.9    70.66   10.1    39.9    65
## 5   California      21198   5114        1.1    71.71   10.3    62.6    20
## 6     Colorado       2541   4884        0.7    72.06    6.8    63.9   166
## 7  Connecticut       3100   5348        1.1    72.48    3.1    56.0   139
## 8     Delaware        579   4809        0.9    70.06    6.2    54.6   103
## 9      Florida       8277   4815        1.3    70.66   10.7    52.6    11
## 10     Georgia       4931   4091        2.0    68.54   13.9    40.6    60
##       Area
## 1    50708
## 2   566432
## 3   113417
## 4    51945
## 5   156361
## 6   103766
## 7     4862
## 8     1982
## 9    54090
## 10   58073
```

(a) The names of the states should be the row names.

```
df8.a <- read.csv("states1.csv",row.names = 1)
head(df8.a,10)
```

```
##             Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## Alabama           3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska             365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona           2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas          2110   3378        1.9    70.66   10.1    39.9    65  51945
## California       21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado          2541   4884        0.7    72.06    6.8    63.9   166 103766
## Connecticut       3100   5348        1.1    72.48    3.1    56.0   139   4862
## Delaware           579   4809        0.9    70.06    6.2    54.6   103   1982
## Florida           8277   4815        1.3    70.66   10.7    52.6    11  54090
## Georgia           4931   4091        2.0    68.54   13.9    40.6    60  58073
```

(b) Check the dimensions of both 'df8' and 'df8.a' data.

```
dim(df8);dim(df8.a)
```

```
## [1] 50  9
```

```
## [1] 50  8
```

Exercise 9:

Read the file 'states2.csv';

```
df9 <- read.csv("states2.csv",sep = ";")
head(df9,10)
```

```
##              X Population Income Illiteracy Life.Exp Murder HS.Grad Frost
## 1      Alabama       3615   3624        2,1    69,05   15,1    41,3    20
## 2       Alaska        365   6315        1,5    69,31   11,3    66,7   152
## 3      Arizona       2212   4530        1,8    70,55    7,8    58,1    15
## 4     Arkansas       2110   3378        1,9    70,66   10,1    39,9    65
## 5   California      21198   5114        1,1    71,71   10,3    62,6    20
## 6     Colorado       2541   4884        0,7    72,06    6,8    63,9   166
## 7  Connecticut       3100   5348        1,1    72,48    3,1      56   139
## 8     Delaware        579   4809        0,9    70,06    6,2    54,6   103
## 9      Florida       8277   4815        1,3    70,66   10,7    52,6    11
## 10     Georgia       4931   4091          2    68,54   13,9    40,6    60
##      Area
## 1   50708
## 2  566432
## 3  113417
## 4   51945
## 5  156361
## 6  103766
## 7    4862
## 8    1982
## 9   54090
## 10  58073
```

The names of the states should be the row names, watch out for the decimal separator and the separator.

```
df9.a <- read.csv("states2.csv",row.names = 1,
                  sep = ";",dec = ",")
head(df9.a,10)
```

```
##            Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
```

```
## Arizona            2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas           2110   3378        1.9    70.66   10.1    39.9    65  51945
## California        21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado           2541   4884        0.7    72.06    6.8    63.9   166 103766
## Connecticut        3100   5348        1.1    72.48    3.1    56.0   139   4862
## Delaware            579   4809        0.9    70.06    6.2    54.6   103   1982
## Florida            8277   4815        1.3    70.66   10.7    52.6    11  54090
## Georgia            4931   4091        2.0    68.54   13.9    40.6    60  58073
```

Exercise 10:

Read the file 'states3.csv';

```
df10 <- read.csv("states3.csv",row.names = 1,
                 sep = ";" ,dec = ",",
                 na.strings = c(NA, "*"))
head(df10,10)
```

```
##                state.division state.area
## Alabama     East South Central      51609
## Alaska                   <NA>     589757
## Arizona              Mountain     113909
## Arkansas    West South Central      53104
## California               <NA>     158693
## Colorado             Mountain     104247
## Connecticut       New England       5009
## Delaware       South Atlantic       2057
## Florida        South Atlantic      58560
## Georgia        South Atlantic         NA
```

Watch out for the same as the last exercise plus the missing values. Add to the previous data set, column-wise.

```
head(cbind(df8.a, df9.a),10)
```

```
##             Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## Alabama           3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska             365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona           2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas          2110   3378        1.9    70.66   10.1    39.9    65  51945
## California       21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado          2541   4884        0.7    72.06    6.8    63.9   166 103766
## Connecticut       3100   5348        1.1    72.48    3.1    56.0   139   4862
## Delaware           579   4809        0.9    70.06    6.2    54.6   103   1982
## Florida           8277   4815        1.3    70.66   10.7    52.6    11  54090
## Georgia           4931   4091        2.0    68.54   13.9    40.6    60  58073
```

```
##             Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## Alabama           3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska             365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona           2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas          2110   3378        1.9    70.66   10.1    39.9    65  51945
## California       21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado          2541   4884        0.7    72.06    6.8    63.9   166 103766
## Connecticut       3100   5348        1.1    72.48    3.1    56.0   139   4862
## Delaware           579   4809        0.9    70.06    6.2    54.6   103   1982
## Florida           8277   4815        1.3    70.66   10.7    52.6    11  54090
## Georgia           4931   4091        2.0    68.54   13.9    40.6    60  58073
```

Exercise 11:

Read 'iris.Rdata' into R.

```
load("iris.Rdata")
```

Writing data files:

Exercise 12:

Using following commands create a data frame and write it to 'data1.txt' file.

```
set.seed(291)
vec1 <- rnorm(15, mean = 5, sd = 2)
vec2 <- sample(100, size = 15)
vec3 <- runif(15)
vec4 <- sample(c("Male","Female"),
               size = 15, replace = T)

data <- data.frame(vec1, vec2, vec3, vec4)
```

```
write.table(data,file = "data1.txt")
```

Exercise 13:

Write the same data set to 'data1.csv' file.

```
write.csv(data, "data1.csv")
```

Add a new column c(15,25,0.5,"Female") to 'data1.csv' file.

```r
# first way
data.new <- rbind(data,c(15,25,0.5,"Female"))
write.csv(data.new, "data1.csv")
```

```r
# second way
write.table(data.frame(15,25,0.5,"Female"), "data1.csv",
            append = TRUE,sep = ",", col.names=F)
```

Exercise 14:

Create an .RData file with same data set, name it 'data1.Rdata'.

```r
save(data, file = "data1.Rdata")
```

Reading Data from Web:

Exercise 15:

Read .csv data from from github.

Go to the github repo with the following URL:

https://github.com/oltuluorcun/Stat291

Read the "data1.csv" file in the repo.

Hint: You have to use the **"Raw"** version of the data set and copy the raw versions URL.

```r
data_github <- read.csv("https://raw.github.com/oltuluorcun/Stat291/main/data1.csv")
data_github
```

```
##     X        vec1 vec2       vec3   vec4
## 1   1  1.25357320   70 0.55901283 Female
## 2   2  2.77134347   88 0.56452973 Female
## 3   3  5.80459322   81 0.55565903 Female
## 4   4  9.67680301    5 0.88892987 Female
## 5   5  4.68679599   46 0.08082044 Female
## 6   6  2.11006254   15 0.82438029   Male
## 7   7  6.19742943   91 0.56203000   Male
## 8   8  6.25596434    4 0.42822813   Male
## 9   9  2.08108099   98 0.89090359   Male
## 10 10  5.11470428   20 0.94826728 Female
## 11 11  6.57811433   47 0.40784030 Female
## 12 12  3.49295708   13 0.80212748   Male
## 13 13  0.06577263   28 0.06934270   Male
## 14 14  6.29646144   69 0.95484243 Female
## 15 15  7.44045290   54 0.80529886   Male
```

```
## 16 16 15.00000000   25 0.50000000 Female
## 17  1 15.00000000   25 0.50000000 Female
```

Exercise 16:

Web scraping with "rvest" package.

Go to the following web page where you will see the current F1 standings for both Drivers and Constructors.

https://www.statsf1.com/en/2021.aspx

Then, using Selector Gagdet, find the 'location' of the tables.

Finally, do the magic and extract the both tables and read into R.

```
library(rvest)
webpage_standing <- "https://www.statsf1.com/en/2021.aspx"
webpage <- read_html(webpage_standing)
calendar_html <- html_nodes(webpage, ".yearclass")

drivers <- html_table(calendar_html[[1]], header = T, na.strings = "-")

drivers <- drivers[-grep("The drivers", drivers$Drivers),]

drivers <- data.frame(drivers)[c(2,25)]

constructors <- html_table(calendar_html[[2]], header = T, na.strings = "-")

constructors <- constructors[-grep("The constr", constructors$Constructors),]

constructors <- data.frame(constructors)[c(2,25)]
```

```
head(drivers,10)
```

```
##        Drivers.1    Pts
## 1  M. VERSTAPPEN 351.50
## 2    L. HAMILTON 343.50
## 3      V. BOTTAS 203.00
## 4       S. PEREZ 190.00
## 5      L. NORRIS 153.00
## 6     C. LECLERC 152.00
## 7       C. SAINZ 145.50
## 8   D. RICCIARDO 105.00
## 9       P. GASLY  92.00
## 10     F. ALONSO  77.00
```

```
constructors
```

```
##           Constructors.1    Pts
## 1               Mercedes 546.50
## 2         Red Bull Honda 541.50
## 3                Ferrari 297.50
## 4       McLaren Mercedes 258.00
## 5         Alpine Renault 137.00
## 6       AlphaTauri Honda 112.00
## 7  Aston Martin Mercedes  77.00
## 8      Williams Mercedes  23.00
## 9     Alfa Romeo Ferrari  11.00
## 10          Haas Ferrari   0.00
```

Check out for further info for usage of 'rvest' package:

https://www.kaggle.com/orcunoltulu1/web-scraping-in-r-rvest-package