

## Homework 1 Predictive Analytics BUAN 6337.006

Submitted by :

Abhishek Arya (Net id : axa220149)

Jitendra Sai Rajesh Motepalli (Net id : mxj230027)

Chandra has Mididuddi (Net id : Cxm220032 )

Your task is to read this data file into R properly

a.

First, examine the raw data file SwineFlu2009.csv using Excel.

Answer a. After exploring the excel file SwineFlu2009 22 unnamed columns are there. Values inside these columns are also unorganized and missing. We know that we have to do a lot of cleaning and filtering. Two date columns are there, 1 character column is there, 1 index column and rest are numeric columns.

b.

Read the data to memory using fread(). Examine the data in Rstudio.

Answer b.

The screenshot displays the RStudio interface with three main panes:

- Script Pane:** Contains R code for reading a CSV file and viewing its structure.

```
4 #b. Read the data into R using fread() from the data.table package.  
5 install.packages("data.table")  
6 library(data.table)  
7  
8 flu_data <- fread("C:/Users/hardi/OneDrive/Desktop/SwinFlu2009.csv")  
9  
10 flu_data  
11  
12  
13  
25:25 (Top Level)
```
- Console Pane:** Shows the output of the R commands.

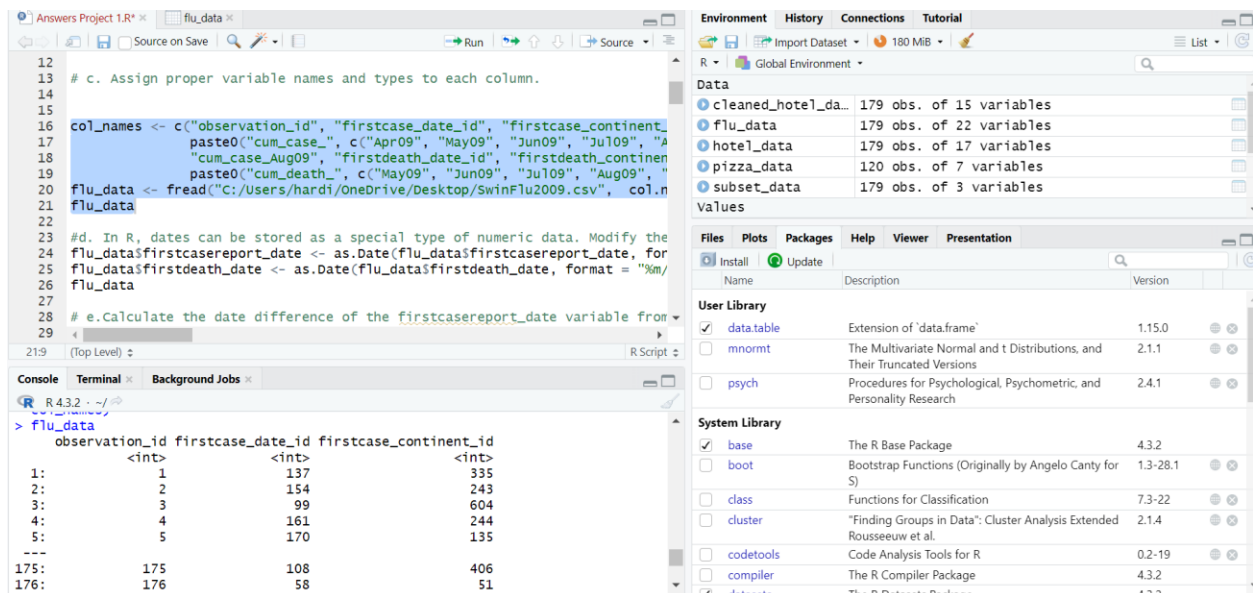
```
> flu_data <- fread("C:/Users/hardi/OneDrive/Desktop/SwinFlu2009.csv")  
> view(flu_data)  
> flu_data  
      V1      V2      V3      V4      V5      V6      V7      V8  
      <int> <int> <int>      <char> <char> <int> <int> <int>  
1:      1     137     335      Afghanistan 7/8/2009      NA      NA      NA  
2:      2     154     243      Albania 7/22/2009      NA      NA      NA  
3:      3      99     604      Algeria 6/22/2009      NA      NA      NA  
4:      4     161     244      Andorra 6/29/2009      NA      NA      NA  
5:      5     170     135      Anguilla 8/5/2009      NA      NA      NA  
---  
175: 175     108     406      Vanuatu 6/24/2009      NA      NA      NA  
176: 176      58      51      Venezuela 6/1/2009      NA      NA      2  
177: 177      63     314      Vietnam 6/1/2009      NA      NA      1  
178: 178      76     318 West Bank and Gaza Strip 6/15/2009      NA      NA      NA  
179: 179      89     322      Yemen 6/17/2009      NA      NA      NA  
      V9      V10      V11      V12      V13      V14      V15      V16      V17      V18      V19  
      <int> <int> <int> <int> <int> <char> <int> <int> <int> <int> <int>  
1:      NA     32      32      99     533 10/30/2009      NA      NA      NA      NA      NA  
2:      NA     32      32      99     533 10/30/2009      NA      NA      NA      NA      NA
```
- Environment Pane:** Lists the objects in the Global Environment.

Object	Obs.	Var.
cleaned_hotel_da...	179	15
flu_data	179	22
hotel_data	179	17
pizza_data	120	7
subset_data	179	3

c. Then, assign the proper variable name to each variable. Make sure that each variable is assigned the correct type – character or numeric. (hint: use `colClasses()` to examine the class of columns)

Answer c. The R-code and screenshot are below:

```
col_names <- c("observation_id", "firstcase_date_id", "firstcase_continent_id", "country", "first
casereport_date",
              paste0("cum_case_", c("Apr09", "May09", "Jun09", "Jul09", "Aug09")),
              "cum_case_Aug09", "firstdeath_date_id", "firstdeath_continent_id", "firstdeath_date",
              paste0("cum_death_", c("May09", "Jun09", "Jul09", "Aug09", "Sep09", "Oct09", "Nov0
9", "Dec09")))
flu_data <- fread("C:/Users/hardi/OneDrive/Desktop/SwinFlu2009.csv", col.names = col_name
s)
flu_data
```



d. In R, dates can be stored as a special type of numeric data. Modify the DATA step to make sure that the dates are read in the correct R date format (not as character). (HINT: Use the correct date type format statements in `as.Date()`, e.g., `format = "%m/%d/%Y"`)

Answer d.

```

flu_data$firstcasereport_date <- as.Date(flu_data$firstcasereport_date, format = "%m/%d/%Y"
)
flu_data$firstdeath_date <- as.Date(flu_data$firstdeath_date, format = "%m/%d/%Y")
flu_data

```

The screenshot shows an R Studio window with a script editor and a console. The script editor contains R code for reading a CSV file and converting dates. The console shows the output of the script, displaying a data frame with columns: country, firstcasereport\_date, and cum\_case\_Apr09.

```

12
13 # c. Assign proper variable names and types to each column.
14
15
16 col_names <- c("observation_id", "firstcase_date_id", "firstcase_continent_
17               paste0("cum_case_", c("Apr09", "May09", "Jun09", "Jul09", "A
18               "cum_case_Aug09", "firstdeath_date_id", "firstdeath_continen
19               paste0("cum_death_", c("May09", "Jun09", "Jul09", "Aug09", "
20 flu_data <- fread("C:/Users/hardi/OneDrive/Desktop/SwinFlu2009.csv", col.n
21 flu_data
22
23 #d. In R, dates can be stored as a special type of numeric data. Modify the
24 flu_data$firstcasereport_date <- as.Date(flu_data$firstcasereport_date, for
25 flu_data$firstdeath_date <- as.Date(flu_data$firstdeath_date, format = "%m/
26 flu_data
27
28
26:9 (Top Level) R Script

```

Console Output:

```

R 4.3.2 ~/>
179:      179      89      322
      country firstcasereport_date cum_case_Apr09
      <char>      <Date>      <int>
1:      Afghanistan      2009-07-08      NA
2:      Albania      2009-07-22      NA
3:      Algeria      2009-06-22      NA
4:      Andorra      2009-06-29      NA
5:      Anguilla      2009-08-05      NA
---
175:      Vanuatu      2009-06-24      NA
176:      Venezuela      2009-06-01      NA
177:      Vietnam      2009-06-01      NA
178: West Bank and Gaza Strip      2009-06-15      NA
179:      Yemen      2009-06-17      NA

```

```

12
13 # c. Assign proper variable names and types to each column.
14
15
16 col_names <- c("observation_id", "firstcase_date_id", "firstcase_continent_
17               paste0("cum_case_", c("Apr09", "May09", "Jun09", "Jul09", "A
18               "cum_case_Aug09", "firstdeath_date_id", "firstdeath_continen
19               paste0("cum_death_", c("May09", "Jun09", "Jul09", "Aug09", "
20 flu_data <- fread("C:/Users/hardi/OneDrive/Desktop/SwinFlu2009.csv", col.n
21 flu_data
22
23 #d. In R, dates can be stored as a special type of numeric data. Modify the
24 flu_data$firstcasereport_date <- as.Date(flu_data$firstcasereport_date, for
25 flu_data$firstdeath_date <- as.Date(flu_data$firstdeath_date, format = "%m/
26 flu_data
27
28
26:9 (Top Level) R Script

```

```

R 4.3.2 ~ /
firstdeath_date cum_death_May09 cum_death_Jun09 cum_death_Jul09
      <Date>          <int>          <int>          <int>
1: 2009-10-30          NA          NA          NA
2: <NA>              NA          NA          NA
3: 2009-11-30          NA          NA          NA
4: <NA>              NA          NA          NA
5: <NA>              NA          NA          NA
---
175: <NA>            NA          NA          NA
176: 2009-07-20      NA          NA          NA
177: 2009-08-05      NA          NA          NA
178: 2009-08-09      NA          NA          NA
179: 2009-08-19      NA          NA          NA

```

e. Calculate the date difference of the first case report\_date variable from the first case report date across the world, which is Apr 24, 2009

Answer e.

```
# Define the world's first case report date
world_first_case_date <- as.Date("2009-04-24")
```

```
# Calculate the date difference for each observation
```

```
flu_data$days_from_first_incidence <- flu_data$firstcasereport_date - world_first_case_date
flu_data
```

The screenshot shows an R Studio window with a script editor and a console. The script editor contains R code for data manipulation. The console shows the output of the code, including the creation of a new variable 'days\_from\_first\_incidence' and the resulting data structure.

```

23 #d. In R, dates can be stored as a special type of numeric data. Modify the
24 flu_data$firstcasereport_date <- as.Date(flu_data$firstcasereport_date, for
25 flu_data$firstdeath_date <- as.Date(flu_data$firstdeath_date, format = "%m/
26 flu_data
27
28 # e. Calculate the date difference of the firstcasereport_date variable from
29 # case report date across the world, which is Apr 24, 2009
30 # Define the world's first case report date
31 world_first_case_date <- as.Date("2009-04-24")
32
33 # Calculate the date difference for each observation
34 flu_data$days_from_first_incidence <- flu_data$firstcasereport_date - world
35 flu_data
36 # f. Subset the columns ("firstcase_date_id", "country") and the answer fro
37
38 # Subset the desired columns

```

The console output shows the following data structure:

```

R 4.3.2 ~ /
178:      NA      1      1      1
179:      NA      1      6     16
      cum_death_Dec09 days_from_first_incidence
      <int>          <difftime>
1:      16          75 days
2:      NA          89 days
3:      3          59 days
4:      NA          66 days
5:      NA         103 days
---
175:      NA          61 days
176:     114          38 days
177:      44          38 days
178:      9          52 days

```

f. Subset the columns ("firstcase\_date\_id", "country") and the answer from the above question 1.e, and save it as the file "SwineFlu2009\_days\_from\_first\_incidence.csv") using fwrite(). (HINT: the new csv file should have three columns)

Answer f.

```

# Subset the desired columns
subset_data <- flu_data[, c("firstcase_date_id", "country", "days_from_first_incidence")]

# Save the subset data as a new CSV file
fwrite(subset_data, "SwineFlu2009_days_from_first_incidence.csv")

file_path <- "C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_days_from_first_incidence.csv"

```

```
# Save the subset data as a new CSV file
fwrite(subset_data, file_path)
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
36 # f.Subset the columns ("firstcase_date_id", "country") and the answer fro
37
38 # Subset the desired columns
39 subset_data <- flu_data[, c("firstcase_date_id", "country", "days_from_firs
40
41 # Save the subset data as a new CSV file
42 fwrite(subset_data, "SwineFlu2009_days_from_first_incidence.csv")
43
44 file_path <- "C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_days_from_firs
45
46 # Save the subset data as a new CSV file
47 fwrite(subset_data, file_path)
48
49
50
51
52
```

The console shows the execution of the code:

```
> # Subset the desired columns
> subset_data <- flu_data[, c("firstcase_date_id", "country", "days_from_firs
>
> # Save the subset data as a new CSV file
> fwrite(subset_data, "SwineFlu2009_days_from_first_incidence.csv")
>
> file_path <- "C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_days_from_firs
>
> # Save the subset data as a new CSV file
> fwrite(subset_data, file_path)
```

The Environment pane shows the following objects:

Object	Class	Attributes
flu_data	data.frame	120 obs. of 7 variables
subset_data	data.frame	179 obs. of 3 variables

The Values pane shows the following values:

col_names	chr [1:22]
file_path	"C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_...
grand_total_247	493.45
world_first_case...	2009-04-24 UTC

# 2.

# a. Examine the raw data file Pizza.csv and read it into R using fread().

Answer 2 a.

```
pizza_data <- fread("C:/Users/hardi/OneDrive/Desktop/Pizza.csv")
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
47 fwrite(subset_data, file_path)
48
49
50
51 # 2.
52 # a. Examine the raw data file Pizza.csv and read it into R using fread().
53 pizza_data <- fread("C:/Users/hardi/OneDrive/Desktop/Pizza.csv")
54 #b. Print the data set (on the Console).
55 print(pizza_data)
56
57 # c. Print the class of each column
58 sapply(pizza_data, class)
59
60 # d. Print the summary statistics of the data using describe() in "psych" p
61 install.packages("psych")
62
```

The console shows the execution of the code:

```
> pizza_data <- fread("C:/Users/hardi/OneDrive/Desktop/Pizza.csv")
> #b. Print the data set (on the Console).
> print(pizza_data)
  SurveyNum Arugula PineNut Squash Shrimp Eggplant
1:    101      1      3      3      NA      NA
2:    102      5      4      2      NA      NA
3:    103      4      2      5      NA      NA
4:    104      5      3      2      NA      NA
5:    105      3      5      5      NA      NA
---
116:   1206      NA      4      1      4      NA
117:   1207      NA      1      1      5      NA
118:   1208      NA      3      1      1      NA
```

The Environment pane shows the following objects:

Object	Class	Attributes
flu_data	data.frame	120 obs. of 7 variables
subset_data	data.frame	179 obs. of 3 variables
pizza_data	data.frame	120 obs. of 6 variables

The Values pane shows the following values:

col_names	chr [1:22]
file_path	"C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_...
grand_total_247	493.45
world_first_case...	2009-04-24 UTC



#b. Print the data set (on the Console).

Answer 2b.

`print(pizza_data)`

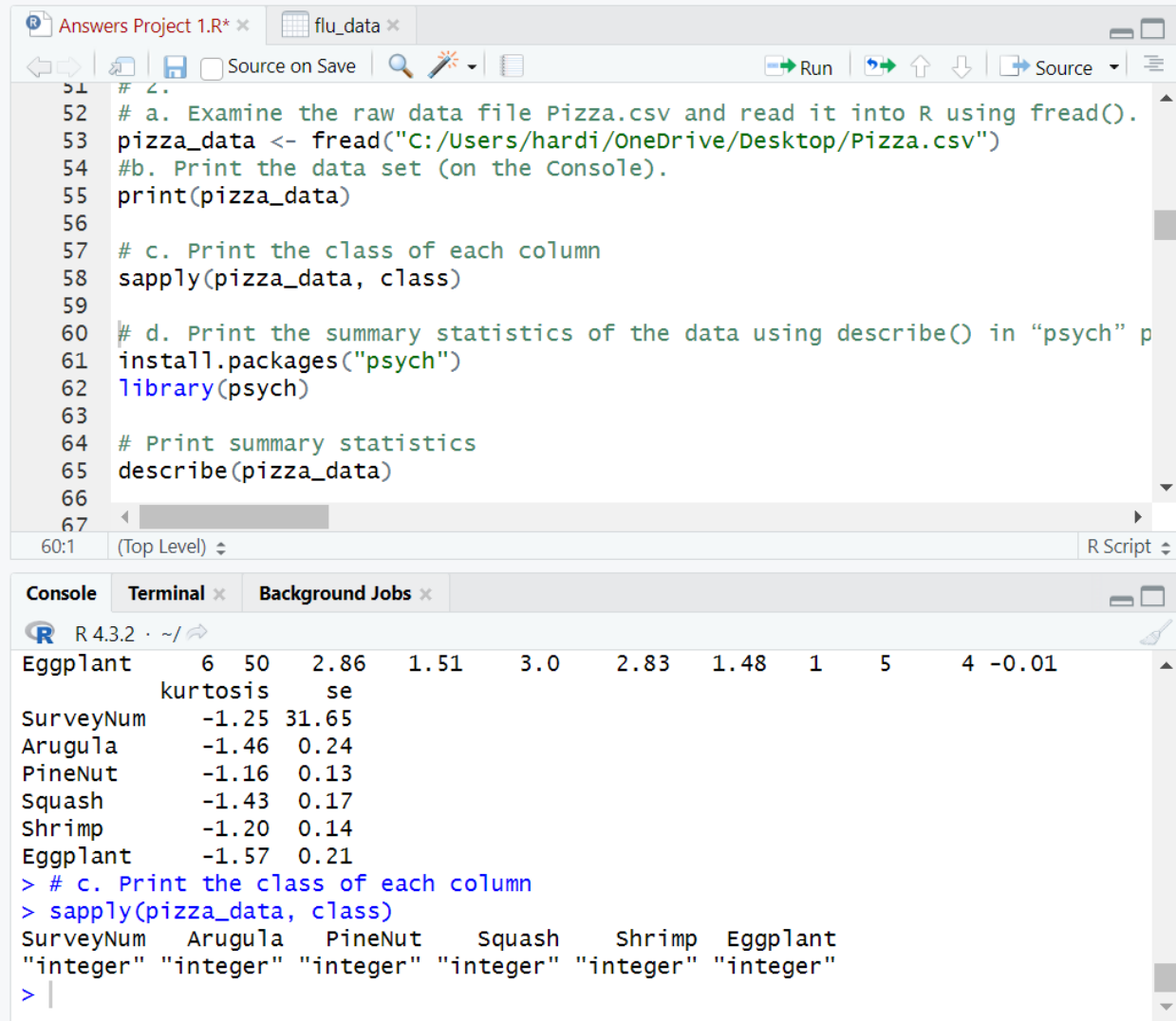
The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for reading a CSV file and printing its contents. The code includes comments and function calls like `fread()`, `print()`, and `sapply()`.
- Console:** Shows the output of the R code, displaying a data frame with columns: `SurveyNum`, `Arugula`, `PineNut`, `Squash`, `Shrimp`, and `Eggplant`. The data is presented in a tabular format with row indices.
- Environment:** Lists the objects in the global environment, including `pizza_data` (120 obs. of 6 variables) and `subset_data` (179 obs. of 3 variables).
- Files:** Shows the file path for the data source: `"C:/Users/hardi/OneDrive/Desktop/SwineFlu2009_..."`.
- Packages:** Lists installed and available packages, including `data.table`, `mnormt`, `psych`, `base`, `boot`, `class`, `cluster`, `codetools`, and `compiler`.

# c. Print the class of each column

Answer 2c.

`sapply(pizza_data, class)`



The screenshot shows an RStudio window with a script editor and a console. The script editor contains R code for reading a CSV file, printing it, checking column classes, and installing the 'psych' package. The console shows the output of these commands, including a data frame of pizza ingredients and their statistics, and the class of each column.

```
# 2.
# a. Examine the raw data file Pizza.csv and read it into R using fread().
pizza_data <- fread("C:/Users/hardi/OneDrive/Desktop/Pizza.csv")
#b. Print the data set (on the Console).
print(pizza_data)

# c. Print the class of each column
sapply(pizza_data, class)

# d. Print the summary statistics of the data using describe() in "psych" p
install.packages("psych")
library(psych)

# Print summary statistics
describe(pizza_data)
```

Console output:

```
R 4.3.2 ~ /
Eggplant      6  50  2.86  1.51  3.0  2.83  1.48  1  5  4 -0.01
      kurtosis  se
SurveyNum    -1.25 31.65
Arugula      -1.46  0.24
PineNut      -1.16  0.13
Squash       -1.43  0.17
Shrimp       -1.20  0.14
Eggplant     -1.57  0.21
> # c. Print the class of each column
> sapply(pizza_data, class)
SurveyNum Arugula PineNut Squash Shrimp Eggplant
"integer" "integer" "integer" "integer" "integer" "integer"
> |
```

# d. Print the summary statistics of the data using describe() in "psych" package.

Answer 2d.

```
install.packages("psych")
```

```
library(psych)
```

```
# Print summary statistics
```

```
describe(pizza_data)
```



The screenshot shows an RStudio window with two tabs: 'Answers Project 1.R\*' and 'flu\_data'. The 'Source' pane contains R code for steps 56 through 71. The 'Console' pane shows the output of the code, including the installation of the 'psych' package and the results of the 'describe()' function.

```

56
57 # c. Print the class of each column
58 sapply(pizza_data, class)
59
60 # d. Print the summary statistics of the data using describe() in "psych" p
61 install.packages("psych")
62 library(psych)
63
64 # Print summary statistics
65 describe(pizza_data)
66
67 # e. Open the raw data file in a simple editor like wordPad and compare the
68 # There may be issues with the column types not being read correctly, leadi
69 # For example, if a numeric column is read as character, it may affect summr
70 #f. Read the same raw data file, Pizza.csv, again. This time, make sure the
71

```

Console Output:

```

> library(psych)
> # Print summary statistics
> describe(pizza_data)

```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
SurveyNum	1	120	655.50	346.66	655.5	655.50	444.78	101	1210	1109	0.00
Arugula	2	40	3.08	1.49	3.0	3.09	1.48	1	5	4	-0.12
PineNut	3	100	3.14	1.29	3.0	3.17	1.48	1	5	4	0.02
Squash	4	80	3.16	1.51	3.0	3.20	2.22	1	5	4	-0.14
Shrimp	5	90	2.97	1.33	3.0	2.96	1.48	1	5	4	-0.03
Eggplant	6	50	2.86	1.51	3.0	2.83	1.48	1	5	4	-0.01
	kurtosis		se								
SurveyNum			-1.25	31.65							
Arugula			-1.46	0.24							
PineNut			-1.16	0.12							

# e. Open the raw data file in a simple editor like WordPad and compare the data values to the output from part b) to make sure that they were read correctly into R. In a comment in your report, identify any problems with the R data set that cannot be resolved using the `fread()`. Explain what is causing the problem. (Hint: You need to make sure the type of each variable is read correctly.)

Answer 2e.

# There may be issues with the column types not being read correctly, leading to incorrect data representation.  
 # For example, if a numeric column is read as character, it may affect summary statistics calculations.

#f. Read the same raw data file, Pizza.csv, again. This time, make sure the issues you've identified in the previous step is resolved.

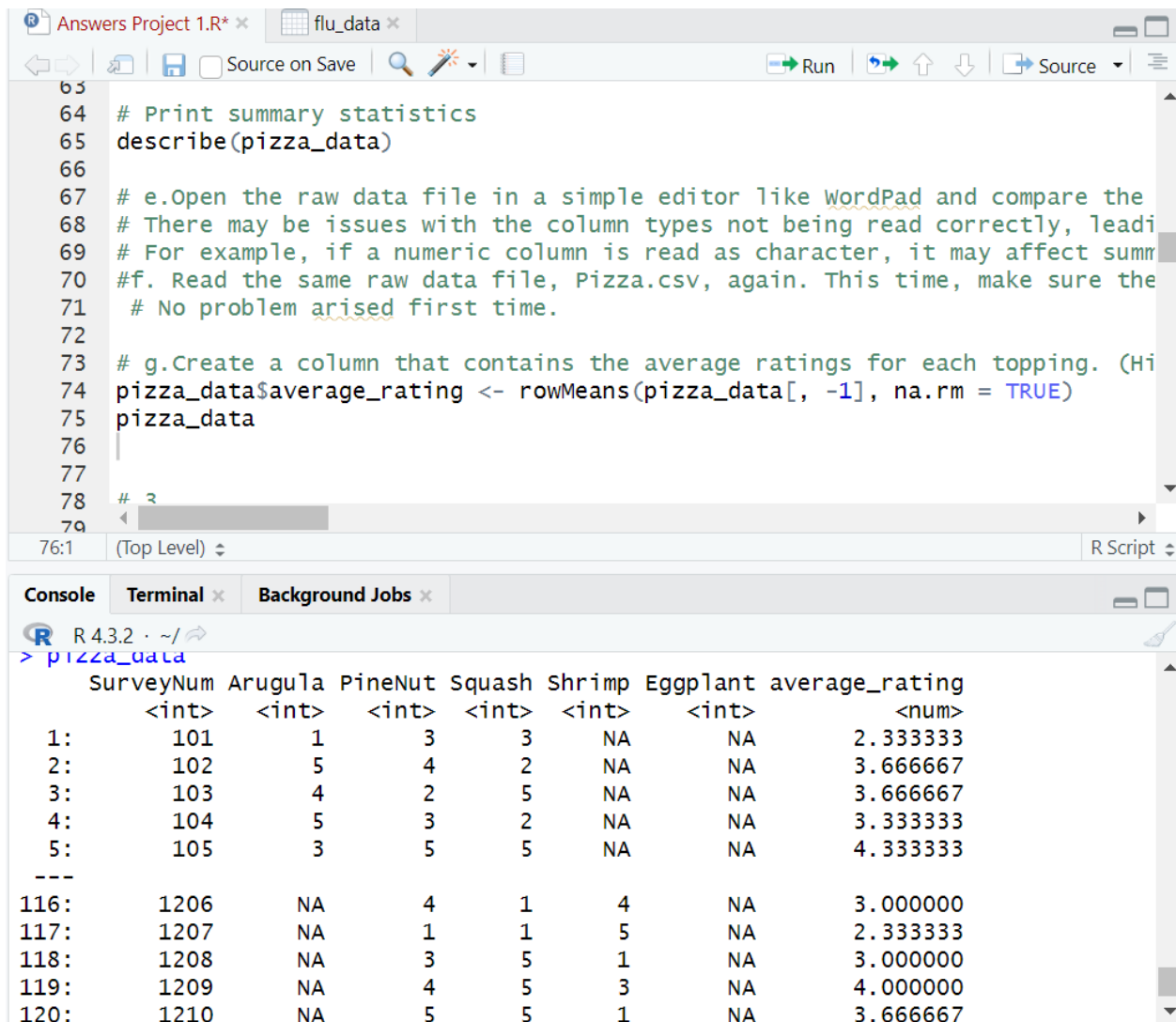
Answer 2 f.

No problem arised first time.

# g. Create a column that contains the average ratings for each topping. (Hint: You need to make sure "NA" entries are not included in the average. They should not be treated as zeros. See the documentation for rowMeans().)

Answer 2g.

```
pizza_data$average_rating <- rowMeans(pizza_data[, -1], na.rm = TRUE)
pizza_data
```



The screenshot shows an RStudio window with a script editor and a console. The script editor contains R code for loading and summarizing data. The console shows the output of the code, including a data frame with columns for survey numbers and ratings for various toppings, plus an average rating column.

```
Answers Project 1.R* x flu_data x
Source on Save Run Source
63
64 # Print summary statistics
65 describe(pizza_data)
66
67 # e.Open the raw data file in a simple editor like WordPad and compare the
68 # There may be issues with the column types not being read correctly, leadi
69 # For example, if a numeric column is read as character, it may affect sumr
70 #f. Read the same raw data file, Pizza.csv, again. This time, make sure the
71 # No problem arised first time.
72
73 # g.Create a column that contains the average ratings for each topping. (Hi
74 pizza_data$average_rating <- rowMeans(pizza_data[, -1], na.rm = TRUE)
75 pizza_data
76
77
78 # 3
79
```

Console Output:

```
R 4.3.2 ~ /
> pizza_data
  SurveyNum Arugula PineNut Squash Shrimp Eggplant average_rating
      <int>  <int>  <int>  <int>  <int>  <int>         <num>
1:      101      1      3      3      NA      NA      2.333333
2:      102      5      4      2      NA      NA      3.666667
3:      103      4      2      5      NA      NA      3.666667
4:      104      5      3      2      NA      NA      3.333333
5:      105      3      5      5      NA      NA      4.333333
---
116:    1206     NA      4      1      4      NA      3.000000
117:    1207     NA      1      1      5      NA      2.333333
118:    1208     NA      3      5      1      NA      3.000000
119:    1209     NA      4      5      3      NA      4.000000
120:    1210     NA      5      5      1      NA      3.666667
```

# 3.

# a. Examine the raw data file Hotel.csv and read it into R using fread(). Is there any "problem" with this data read? Explain.

Answer 3a.

Warning message:

In fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv") :

Stopped early on line 4. Expected 11 fields but found 12. Consider fill=TRUE and comment.char  
=. First discarded non-empty line: <<220,5,2,3,2014,2,12,2014,YES,2,Basic w/view,155>>

The warning message indicates that there was an inconsistency in the number of fields detected by fread() on line 4 of the "Hotel.csv" file. It expected 11 fields but found 12.

To address this issue, you can use the fill = TRUE parameter in the fread() function, which allows fread() to fill missing values with NA when the number of fields is inconsistent across lines.

Additionally, you can specify the verbose = TRUE parameter to handle comments if necessary.

```
hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv")
```

```
hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv", fill= TRUE, verbose = TRUE)
```

```
hotel_data
```

Answers Project 1.R\* x

Source on Save

Run

Source

```
73 # g.Create a column that contains the average ratings for each topping. (Hi
74 pizza_data$average_rating <- rowMeans(pizza_data[, -1], na.rm = TRUE)
75 pizza_data
76
77
78 # 3.
79 # a. Examine the raw data file Hotel.csv and read it into R using fread().
80 hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv")
81 hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv", fill= TRUE
82 hotel_data
83
84 # b. Assign the column names for room number and number of guests first. Fo
85 # Assign column names for room number and number of guests
86 # Assign column names for specific columns
87 names(hotel_data)[1] <- "RoomNumber"
88
```

81:1 (Top Level) R Script

Console

Terminal x

Background Jobs x

R 4.3.2 · ~/

```
177: Basic no view      75
178:           Suite    255
179:           155      NA
> # 3.
> # a. Examine the raw data file Hotel.csv and read it into R using fread(). Is t
here any "problem" with this data read? Explain.
> hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv")
Warning message:
In fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv") :
  Stopped early on line 4. Expected 11 fields but found 12. Consider fill=TRUE an
d comment.char=. First discarded non-empty line: <<220,5,2,3,2014,2,12,2014,YES,
2,Basic w/view,155>>
>
```

Answers Project 1.R\* x

Source on Save Run Source

```

73 # g.Create a column that contains the average ratings for each topping. (Hi
74 pizza_data$average_rating <- rowMeans(pizza_data[, -1], na.rm = TRUE)
75 pizza_data
76
77
78 # 3.
79 # a. Examine the raw data file Hotel.csv and read it into R using fread().
80 hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv")
81 hotel_data <- fread("C:/Users/hardi/OneDrive/Desktop/Hotel.csv", fill= TRUE
82 hotel_data
83
84 # b. Assign the column names for room number and number of guests first. Fo
85 # Assign column names for room number and number of guests
86 # Assign column names for specific columns
87 names(hotel_data)[1] <- "RoomNumber"
88

```

82:11 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · ~/

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<char>	<char>
1:	211	3	2	7	2014	2	11	2014	NO	Deluxe Suite
2:	214	2	2	2	2014	2	12	2014	NO	Basic no view
3:	216	4	2	2	2014	2	13	2014	NO	Suite
4:	220	5	2	3	2014	2	12	2014	YES	2
5:	221	3	2	3	2014	2	12	2014	NO	Luxury
---										
175:	1276	2	2	1	2014	2	11	2014	YES	3
176:	1285	5	1	31	2014	2	11	2014	NO	Deluxe Suite
177:	1291	1	2	6	2014	2	10	2014	YES	4
178:	1294	6	2	8	2014	2	13	2014	YES	5
179:	1298	7	2	7	2014	2	11	2014	NO	Basic w/view

# b. Assign the column names for room number and number of guests first. For other column names, you should assign them as you answer the remaining questions.

Answer 3b.

# Assign column names for room number and number of guests

# Assign column names for specific columns

names(hotel\_data)[1] <- "RoomNumber"

names(hotel\_data)[2] <- "NumberOfGuests"

```
Answers Project 1.R* x
Source on Save
Run
Source

83
84 # b. Assign the column names for room number and number of guests first. For
85 # Assign column names for room number and number of guests
86 # Assign column names for specific columns
87 names(hotel_data)[1] <- "RoomNumber"
88 names(hotel_data)[2] <- "NumberOfGuests"
89 hotel_data
90 # c. Create date variables for the check-in and check-out dates, and format
91 # Read the data without specifying column names
92 hotel_data
93 names(hotel_data)[3] <- "monthin"
94 names(hotel_data)[4] <- "dayin"
95 names(hotel_data)[5] <- "yearin"
96 names(hotel_data)[6] <- "monthout"
97 names(hotel_data)[7] <- "dayout"
98

89:11 (Top Level) R Script

Console Terminal x Background Jobs x
R 4.3.2 ~ /
> hotel_data
  RoomNumber NumberOfGuests monthin dayin yearin monthout dayout yearout
      <int>         <int>   <int> <int>   <int>   <int>   <int>   <int>
1:         211             3       2     7    2014       2      11    2014
2:         214             2       2     2    2014       2      12    2014
3:         216             4       2     2    2014       2      13    2014
4:         220             5       2     3    2014       2      12    2014
5:         221             3       2     3    2014       2      12    2014
---
175:        1276             2       2     1    2014       2      11    2014
176:        1285             5       1    31    2014       2      11    2014
177:        1291             1       2     6    2014       2      10    2014
178:        1294             6       2     8    2014       2      13    2014
179:        1298             7       2     7    2014       2      11    2014
```

# c. Create date variables for the check-in and check-out dates, and format them to display as readable dates.

Answer 3c.

# Read the data without specifying column names

hotel\_data

names(hotel\_data)[3] <- "monthin"

names(hotel\_data)[4] <- "dayin"

names(hotel\_data)[5] <- "yearin"

names(hotel\_data)[6] <- "monthout"

names(hotel\_data)[7] <- "dayout"

names(hotel\_data)[8] <- "yearout"

# Combine check-in date components into a single date variable



hotel\_data

```
hotel_data$checkindate <- as.Date(with(hotel_data, paste(monthin, dayin, yearin, sep="-")), "%m-%d-%Y")
```

```
hotel_data$checkoutdate <- as.Date(with(hotel_data, paste(monthout, dayout, yearout, sep="-")), "%m-%d-%Y")
```

hotel\_data

The screenshot shows the R Studio interface. The top pane displays R code for processing hotel data. The bottom pane shows the console output, which includes a summary of the first few rows of the data and a detailed view of the last row (row 175).

```
109:1 names(hotel_data)[8] <- "yearout"
110:1 # Combine check-in date components into a single date variable
111:1 hotel_data
112:1 hotel_data$checkindate <- as.Date(with(hotel_data, paste(monthin, dayin, ye
113:1 hotel_data$checkoutdate <- as.Date(with(hotel_data, paste(monthout, dayout,
114:1 hotel_data
115:1 #d.Using the data.table syntax, create a column of days of internet use. If
116:1 names(hotel_data)[9] <- "InternetUse"
```

Console output:

```
175:      1276      2      2      1 2014      2      11 2014
176:      1285      5      1     31 2014      2      11 2014
177:      1291      1      2      6 2014      2      10 2014
178:      1294      6      2      8 2014      2      13 2014
179:      1298      7      2      7 2014      2      11 2014
```

	V9	V10	V11	V12	checkindate	checkoutdate
	<char>	<char>	<char>	<int>	<Date>	<Date>
1:	NO	Deluxe Suite	295	NA	2014-02-07	2014-02-11
2:	NO	Basic no view	75	NA	2014-02-02	2014-02-12
3:	NO	Suite	255	NA	2014-02-02	2014-02-13
4:	YES	2 Basic w/view	155		2014-02-03	2014-02-12
5:	NO	Luxury	195	NA	2014-02-03	2014-02-12
---						
175:	YES	3 Basic w/view	155		2014-02-01	2014-02-11

#d.Using the data.table syntax, create a column of days of internet use. If the guest did not use the internet, assign "0". Check the class of the column you created and coerce the variable type

to “numeric” as necessary. (Hint. Days of internet use is recorded only when the use of wireless internet service is YES. See the documentation for `as.numeric()` and `as.character()`)

Answer 3d.

```
names(hotel_data)[9] <- "InternetUse"
```

```
hotel_data[, DaysOfInternetUse := 0]
```

```
# Assuming the column names for internet use and days of internet use are 'InternetUse' and 'DaysOfInternetUse' respectively
```

```
hotel_data[, DaysOfInternetUse := ifelse(InternetUse == "YES", as.numeric(as.character(V10)), 0)]
```

```
# Check the class of the column
```

```
print(class(hotel_data$DaysOfInternetUse))
```

```
# Coerce to numeric if necessary
```

```
hotel_data$DaysOfInternetUse <- as.numeric(hotel_data$DaysOfInternetUse)
```

```
hotel_data
```

```

Answers Project 1.R* x
Source on Save Run Source
109 #d.Using the data.table syntax, create a column of days of internet use. It
110
111 names(hotel_data)[9] <- "InternetUse"
112
113 hotel_data[, DaysOfInternetUse := 0]
114
115 # Assuming the column names for internet use and days of internet use are '
116 hotel_data[, DaysOfInternetUse := ifelse(InternetUse == "YES", as.numeric(a
117
118 # Check the class of the column
119 print(class(hotel_data$DaysOfInternetUse))
120
121 # Coerce to numeric if necessary
122 hotel_data$DaysOfInternetUse <- as.numeric(hotel_data$DaysOfInternetUse)
123 hotel_data
124 # e.Using the data.table syntax, create a column of room type. (Again, use
125
124:25 (Top Level) R Script

```

Console Terminal Background Jobs							
R 4.3.2 · ~/							
179:	1298	7	2	7	2014	2	11
	InternetUse	V10		V11	V12	checkindate	checkoutdate
	<char>	<char>		<char>	<int>	<Date>	<Date>
1:	NO	Deluxe Suite		295	NA	2014-02-07	2014-02-11
2:	NO	Basic no view		75	NA	2014-02-02	2014-02-12
3:	NO	Suite		255	NA	2014-02-02	2014-02-13
4:	YES	2	Basic w/view	155	2014-02-03	2014-02-12	
5:	NO	Luxury		195	NA	2014-02-03	2014-02-12
---							
175:	YES	3	Basic w/view	155	2014-02-01	2014-02-11	
176:	NO	Deluxe Suite		295	NA	2014-01-31	2014-02-11
177:	YES	4	Basic no view	75	2014-02-06	2014-02-10	
178:	YES	5	suite	255	2014-02-08	2014-02-13	

```

Answers Project 1.R* x
Source on Save Run Source
109 #d.Using the data.table syntax, create a column of days of internet use. It
110
111 names(hotel_data)[9] <- "InternetUse"
112
113 hotel_data[, DaysOfInternetUse := 0]
114
115 # Assuming the column names for internet use and days of internet use are '
116 hotel_data[, DaysOfInternetUse := ifelse(InternetUse == "YES", as.numeric(a
117
118 # Check the class of the column
119 print(class(hotel_data$DaysOfInternetUse))
120
121 # Coerce to numeric if necessary
122 hotel_data$DaysOfInternetUse <- as.numeric(hotel_data$DaysOfInternetUse)
123 hotel_data
124 # e.Using the data.table syntax, create a column of room type. (Again, use
125
124:25 (Top Level) R Script

```

```

Console Terminal x Background Jobs x
R 4.3.2 ~/
177: YES 4 Basic no view 75 2014-02-06 2014-02-10
178: YES 5 Suite 255 2014-02-08 2014-02-13
179: NO Basic w/view 155 NA 2014-02-07 2014-02-11
DaysOfInternetUse
<num>
1: 0
2: 0
3: 0
4: 2
5: 0
---
175: 3
176: 0
177: 4

```

# e.Using the data.table syntax, create a column of room type. (Again, use the hint from the above)

Answer 3e.

```
hotel_data[, roomtype := ""]
```

```
hotel_data[, roomtype := gsub("[0-9]", "", paste(V10, V11))]
hotel_data
```

```

Answers Project 1.R* x
Source on Save Run Source
119 print(class(hotel_data$DaysOfInternetUse))
120
121 # Coerce to numeric if necessary
122 hotel_data$DaysOfInternetUse <- as.numeric(hotel_data$DaysOfInternetUse)
123 hotel_data
124 # e. Using the data.table syntax, create a column of room type. (Again, use
125 hotel_data[, roomtype := ""]
126
127 hotel_data[, roomtype := gsub("[0-9]", "", paste(V10, V11))]
128 hotel_data
129 # f. Using the data.table syntax, create a column of room rate. Check the c
130 hotel_data[, roomrate := 0]
131
132 hotel_data[, roomrate := as.numeric(gsub("[^0-9.]", "", paste(V11, V12)))]
133
134 names(hotel_data)[15] <- "number_of_days_of_Internet_use"
135
128:11 (Top Level) R Script

```

```

R 4.3.2 ~ /
177: YES 4 Basic no view 75 2014-02-06 2014-02-10
178: YES 5 Suite 255 2014-02-08 2014-02-13
179: NO Basic w/view 155 NA 2014-02-07 2014-02-11
      DaysOfInternetUse roomtype
      <num> <char>
1: 0 Deluxe Suite
2: 0 Basic no view
3: 0 Suite
4: 2 Basic w/view
5: 0 Luxury
---
175: 3 Basic w/view
176: 0 Deluxe Suite
177: 4 Basic no view

```

# f. Using the data.table syntax, create a column of room rate. Check the class of the column you created and coerce the variable type to “numeric” as necessary. (Again, use the hint from the above)

Answer 3f.

```
hotel_data[, roomrate := 0]
```

```
hotel_data[, roomrate := as.numeric(gsub("[^0-9.]", "", paste(V11, V12)))]
```

```
names(hotel_data)[15] <- "number_of_days_of_Internet_use"
names(hotel_data)[9] <- "use_of_wireless_Internet_service"
hotel_data
```

```

Answers Project 1.R* x
Source on Save Run
125 hotel_data[, roomtype := ]
126
127 hotel_data[, roomtype := gsub("[0-9]", "", paste(V10, V11))]
128 hotel_data
129 # f. Using the data.table syntax, create a column of room rate. Check the c
130 hotel_data[, roomrate := 0]
131
132 hotel_data[, roomrate := as.numeric(gsub("[^0-9.]", "", paste(V11, V12)))]
133
134 names(hotel_data)[15] <- "number_of_days_of_Internet_use"
135 names(hotel_data)[9] <- "use_of_wireless_Internet_service"
136 hotel_data
137
138
139 # g.Subset the cleaned variables only and create a new data.table: room num
140 # Create a new data table with selected columns
141
136:11 (Top Level) R Script

```

```

R 4.3.2 · ~/
176: 2014-01-31 2014-02-11 0 Deluxe Suite
177: 2014-02-06 2014-02-10 4 Basic no view
178: 2014-02-08 2014-02-13 5 Suite
179: 2014-02-07 2014-02-11 0 Basic w/view
roomrate
<num>
1: 295
2: 75
3: 255
4: 155
5: 195
---
175: 155
176: 295

```

# g.Subset the cleaned variables only and create a new data.table: room number, number of guests, check-in date, check-out date, use of wireless Internet service, number of days of Internet use, room type, and room rate.

Answer 3g.

# Create a new data.table with selected columns



```
cleaned_hotel_data <- hotel_data[, .(RoomNumber, NumberOfGuests, checkindate, checkoutedate, use_of_wireless_Internet_service, number_of_days_of_Internet_use, roomtype, roomrate)]
cleaned_hotel_data
```

The screenshot shows the RStudio environment. The top pane is the 'Source' editor, displaying R code for cleaning hotel data. The bottom pane is the 'Console', showing the execution of the code and the resulting data structure.

**Source Editor Code:**

```
135 names(hotel_data)[9] <- "use_of_wireless_Internet_service"
136 hotel_data
137
138
139 # g.Subset the cleaned variables only and create a new data.table: room number, number of guests, check-in date, c
140 # Create a new data.table with selected columns
141 cleaned_hotel_data <- hotel_data[, .(RoomNumber, NumberOfGuests, checkindate, checkoutedate, use_of_wireless_Intern
142 cleaned_hotel_data
143
144
145
146 # Assuming cleaned_hotel_data is your data.table and you have the necessary cleaned columns: RoomRate, CheckInDate
147 # h. Create a variable that calculates the subtotal as the room rate times the number of days in the stay, plus a
148 # Calculate the number of days in the stay
149 cleaned_hotel_data[, DaysOfStay := as.numeric(difftime(checkoutedate, checkindate, units = "days"))]
150
151
152 (Top Level)
```

**Console Output:**

```
> cleaned_hotel_data <- hotel_data[, .(RoomNumber, NumberOfGuests, checkindate, checkoutedate, use_of_wireless_Internet_s
service, number_of_days_of_Internet_use, roomtype, roomrate)]
> cleaned_hotel_data
   RoomNumber NumberOfGuests checkindate checkoutedate
   <int>      <int>      <Date>      <Date>
1:      211           3    2014-02-07    2014-02-11
2:      214           2    2014-02-02    2014-02-12
3:      216           4    2014-02-02    2014-02-13
4:      220           5    2014-02-03    2014-02-12
5:      221           3    2014-02-03    2014-02-12
---
175:     1276           2    2014-02-01    2014-02-11
176:     1285           5    2014-01-31    2014-02-11
177:     1291           1    2014-02-06    2014-02-10
```

Answers Project 1.R\* x

```

135 names(hotel_data)[9] <-"use_of_wireless_internet_service"
136 hotel_data
137
138
139 # g.Subset the cleaned variables only and create a new data.table: room number, number of guests, check-in date, c
140 # Create a new data.table with selected columns
141 cleaned_hotel_data <- hotel_data[, .(RoomNumber, NumberOfGuests, checkindate, checkoutdate, use_of_wireless_Intern
142 cleaned_hotel_data
143
144
145
146 # Assuming cleaned_hotel_data is your data.table and you have the necessary cleaned columns: RoomRate, CheckInDate
147 # h. Create a variable that calculates the subtotal as the room rate times the number of days in the stay, plus a
148 # Calculate the number of days in the stay
149 cleaned_hotel_data[, DaysofStay := as.numeric(difftime(checkoutdate, checkindate, units = "days"))]
150
151
142:19 (Top Level)
R Script

```

Console Terminal Background Jobs

R 4.3.2 · ~/

```

179:      1298      7 2014-02-07 2014-02-11
      use_of_wireless_internet_service number_of_days_of_internet_use
      <char> <num>
1:      NO      0
2:      NO      0
3:      NO      0
4:      YES      2
5:      NO      0
---
175:      YES      3
176:      NO      0
177:      YES      4
178:      YES      5
179:      NO      0

```

Answers Project 1.R\* x

```

135 names(hotel_data)[9] <-"use_of_wireless_internet_service"
136 hotel_data
137
138
139 # g.Subset the cleaned variables only and create a new data.table: room number, number of guests, check-in date, c
140 # Create a new data.table with selected columns
141 cleaned_hotel_data <- hotel_data[, .(RoomNumber, NumberOfGuests, checkindate, checkoutdate, use_of_wireless_Intern
142 cleaned_hotel_data
143
144
145
146 # Assuming cleaned_hotel_data is your data.table and you have the necessary cleaned columns: RoomRate, CheckInDate
147 # h. Create a variable that calculates the subtotal as the room rate times the number of days in the stay, plus a
148 # Calculate the number of days in the stay
149 cleaned_hotel_data[, DaysofStay := as.numeric(difftime(checkoutdate, checkindate, units = "days"))]
150
151
142:19 (Top Level)
R Script

```

Console Terminal Background Jobs

R 4.3.2 · ~/

```

177:      YES      4
178:      YES      5
179:      NO      0
      roomtype roomrate
      <char> <num>
1: Deluxe Suite      295
2: Basic no view      75
3: Suite      255
4: Basic w/view      155
5: Luxury      195
---
175: Basic w/view      155
176: Deluxe Suite      295
177: Basic no view      75

```

# Assuming cleaned\_hotel\_data is your data.table and you have the necessary cleaned columns : RoomRate, CheckInDate, CheckOutDate, NumberOfGuests, and DaysOfInternetUse

# h. Create a variable that calculates the subtotal as the room rate times the number of days in the stay, plus a per person rate (\$10 per day for each person beyond one guest), plus an Internet service fee (\$9.95 for a one-time activation and \$5.95 per day of use).

Answer 3h .

# Calculate the number of days in the stay

```
cleaned_hotel_data[, DaysOfStay := as.numeric(difftime(checkoutdate, checkindate, units = "days"))]
```

# Calculate the total room rate for the stay

```
cleaned_hotel_data[, TotalRoomRate := roomrate * DaysOfStay]
```

# Calculate the additional charges for extra guests

```
cleaned_hotel_data[, ExtraGuestCharges := 10 * (NumberOfGuests - 1) * DaysOfStay]
```

# Calculate the internet service fee

```
cleaned_hotel_data[, InternetServiceFee := 9.95 + 5.95 * number_of_days_of_Internet_use]
```

# Calculate the subtotal

```
cleaned_hotel_data[, Subtotal := TotalRoomRate + ExtraGuestCharges + InternetServiceFee]  
cleaned_hotel_data
```

```

152 cleaned_hotel_data[, TotalRoomRate := roomrate * DaysOfStay]
153
154 # Calculate the additional charges for extra guests
155 cleaned_hotel_data[, ExtraGuestCharges := 10 * (NumberOfGuests - 1) * DaysOfStay]
156
157 # Calculate the internet service fee
158 cleaned_hotel_data[, InternetServiceFee := 9.95 + 5.95 * number_of_days_of_Internet_us
159
160 # Calculate the subtotal
161 cleaned_hotel_data[, Subtotal := TotalRoomRate + ExtraGuestCharges + InternetServiceFee
162 cleaned_hotel_data
163
164
165 # Assuming cleaned_hotel_data is your data.table and you have the Subtotal column
166 # i. Create a variable that calculates the grand total as the subtotal plus sales tax
167

```

162:19 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 ~ /

	ExtraGuestCharges	InternetServiceFee	Subtotal
	<num>	<num>	<num>
1:	80	9.95	1269.95
2:	100	9.95	859.95
3:	330	9.95	3144.95
4:	360	21.85	1776.85
5:	180	9.95	1944.95
---			
175:	100	27.80	1677.80
176:	440	9.95	3694.95
177:	0	33.75	333.75
178:	250	39.70	1564.70
179:	240	9.95	869.95

# Assuming cleaned\_hotel\_data is your data.table and you have the Subtotal column  
 # i. Create a variable that calculates the grand total as the subtotal plus sales tax at 8.75%. The result should be rounded to two decimal places.

Answer 3i.

# Calculate the sales tax

cleaned\_hotel\_data[, SalesTax := Subtotal \* 0.0875]

# Calculate the grand total

cleaned\_hotel\_data[, GrandTotal := Subtotal + SalesTax]

# Round the grand total to two decimal places

cleaned\_hotel\_data[, GrandTotal := round(GrandTotal, 2)]

cleaned\_hotel\_data

```

Answers Project 1.R* x
Source on Save Run Source
163
164
165 # Assuming cleaned_hotel_data is your data.table and you have the Subtotal column
166 # i. Create a variable that calculates the grand total as the subtotal plus sales tax
167 # Calculate the sales tax
168 cleaned_hotel_data[, SalesTax := Subtotal * 0.0875]
169
170 # Calculate the grand total
171 cleaned_hotel_data[, GrandTotal := Subtotal + SalesTax]
172
173 # Round the grand total to two decimal places
174 cleaned_hotel_data[, GrandTotal := round(GrandTotal, 2)]
175 cleaned_hotel_data
176
177
178 # j. View the resulting data set. In a comment in your report, state the value for the
179
175:19 (Top Level) R Script

```

```

R 4.3.2 ~ /
178:      5 Suite 255 5 12/5
179:      0 Basic w/view 155 4 620
      ExtraGuestCharges InternetServiceFee Subtotal SalesTax GrandTotal
      <num> <num> <num> <num> <num>
1:      80 9.95 1269.95 111.12063 1381.07
2:     100 9.95 859.95 75.24563 935.20
3:     330 9.95 3144.95 275.18312 3420.13
4:     360 21.85 1776.85 155.47437 1932.32
5:     180 9.95 1944.95 170.18312 2115.13
---
175:     100 27.80 1677.80 146.80750 1824.61
176:     440 9.95 3694.95 323.30812 4018.26
177:      0 33.75 333.75 29.20312 362.95
178:     250 39.70 1564.70 136.91125 1701.61

```

# j. View the resulting data set. In a comment in your report, state the value for the grand total for room 247, checked in on Feb. 7th, 2014.

Answer 3j.

# Calculate the grand total for room 247, checked in on Feb. 7th, 2014

```
grand_total_247 <- cleaned_hotel_data[RoomNumber == 247 & checkindate == "2014-02-07"]$GrandTotal
```

# Print the grand total for room 247

```
cat("Grand total for room 247, checked in on Feb. 7th, 2014:", grand_total_247, "\n")
```

Answers Project 1.R\*

Source on Save Run

```
171 cleaned_hotel_data[, GrandTotal := Subtotal + SalesTax]
172
173 # Round the grand total to two decimal places
174 cleaned_hotel_data[, GrandTotal := round(GrandTotal, 2)]
175 cleaned_hotel_data
176
177
178 # j. View the resulting data set. In a comment in your report, state the value for the
179 # Calculate the grand total for room 247, checked in on Feb. 7th, 2014
180 grand_total_247 <- cleaned_hotel_data[RoomNumber == 247 & checkindate == "2014-02-07"]
181
182 # Print the grand total for room 247
183 cat("Grand total for room 247, checked in on Feb. 7th, 2014:", grand_total_247, "\n")
184
185
```

184:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.2 · ~/

```
177:      0      33.75  333.75  29.20312   362.95
178:    250     39.70 1564.70 136.91125  1701.61
179:    240      9.95  869.95  76.12063   946.07
> # j. View the resulting data set. In a comment in your report, state the value for the grand
total for room 247, checked in on Feb. 7th, 2014.
> # Calculate the grand total for room 247, checked in on Feb. 7th, 2014
> grand_total_247 <- cleaned_hotel_data[RoomNumber == 247 & checkindate == "2014-02-07"]$GrandTotal
>
> # Print the grand total for room 247
> cat("Grand total for room 247, checked in on Feb. 7th, 2014:", grand_total_247, "\n")
Grand total for room 247, checked in on Feb. 7th, 2014: 493.45
>
```