HW1_2023_HXD220000_MXB220061_SXV220020

2023-02-13

Authors : Mankirat Singh Bharma MXB220061 - Harikrishna Dev HXD220000 - Sarthak Vajpayee SXV220020

Loading required libraries and cleaning environment

```
rm(list = ls())
demo = T
require(psych)

## Loading required package: psych

require(data.table)

## Loading required package: data.table
```

Section - 1

1. Swine Flu problem

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

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

```
SwineFlu <- fread("SwinFlu2009.csv",
                  na.strings = c("NA", ""),
                  sep = "auto",
                  stringsAsFactors = FALSE,
                  data.table = TRUE)
str(SwineFlu)

## Classes 'data.table' and 'data.frame':   179 obs. of  22 variables:
##  $ V1 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ V2 : int  137 154 99 161 170 101 28 122 29 9 ...
##  $ V3 : int  335 243 604 244 135 122 503 124 402 203 ...
##  $ V4 : chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
##  $ V5 : chr  "7/8/2009" "7/22/2009" "6/22/2009" "6/29/2009" ...
##  $ V6 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V7 : int  NA NA NA NA NA NA NA NA NA 1 ...
##  $ V8 : int  NA NA NA NA NA NA 100 NA 297 1 ...
##  $ V9 : int  NA NA 2 NA NA 2 1587 NA 4090 15 ...
##  $ V10: int  32 3 16 1 NA 3 3056 13 22109 153 ...
##  $ V11: int  32 13 19 1 1 4 5710 13 24949 192 ...
##  $ V12: int  99 NA 123 NA NA NA 9 NA 11 100 ...
##  $ V13: int  533 NA 613 NA NA NA 203 NA 401 324 ...
##  $ V14: chr  "10/30/2009" NA "11/30/2009" NA ...
##  $ V15: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V16: int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ V17: int  NA NA NA NA NA NA 26 NA 7 NA ...
##  $ V18: int  NA NA NA NA NA NA 165 NA 67 NA ...
##  $ V19: int  NA NA NA NA NA NA 465 NA 155 NA ...
##  $ V20: int  NA NA NA NA NA NA 538 NA 180 NA ...
##  $ V21: int  1 NA NA NA NA NA 593 NA 186 NA ...
##  $ V22: int  16 NA 3 NA NA NA 613 NA 190 3 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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)

```
colClasses = c("observation_id","firstcase_date_id",
               "firstcase_continent_id","country",
               "firstcasereport_date","cum_case_April",

"cum_case_May","cum_case_June","cum_case_July","cum_case_August",

"cum_case_Aug09","firstdeath_date_id","firstdeath_continent_id",

"firstdeath_date","cum_death_May","cum_death_June","cum_death_July",

"cum_death_August","cum_death_September","cum_death_October","cum_death_Novem
ber","cum_death_December")

colnames(SwineFlu) <- colClasses

if(demo) {str(SwineFlu)
  summary(SwineFlu)}
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  22 variables:
##  $ observation_id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ firstcase_date_id      : int  137 154 99 161 170 101 28 122 29 9 ...
##  $ firstcase_continent_id : int  335 243 604 244 135 122 503 124 402 203
...
##  $ country                : chr  "Afghanistan" "Albania" "Algeria"
"Andorra" ...
##  $ firstcasereport_date   : chr  "7/8/2009" "7/22/2009" "6/22/2009"
"6/29/2009" ...
##  $ cum_case_April         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_case_May           : int  NA NA NA NA NA NA NA NA NA 1 ...
##  $ cum_case_June          : int  NA NA NA NA NA NA 100 NA 297 1 ...
##  $ cum_case_July          : int  NA NA 2 NA NA 2 1587 NA 4090 15 ...
##  $ cum_case_August        : int  32 3 16 1 NA 3 3056 13 22109 153 ...
##  $ cum_case_Aug09         : int  32 13 19 1 1 4 5710 13 24949 192 ...
##  $ firstdeath_date_id     : int  99 NA 123 NA NA NA 9 NA 11 100 ...
##  $ firstdeath_continent_id: int  533 NA 613 NA NA NA 203 NA 401 324 ...
##  $ firstdeath_date        : chr  "10/30/2009" NA "11/30/2009" NA ...
##  $ cum_death_May          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_June         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_July         : int  NA NA NA NA NA NA 26 NA 7 NA ...
```

```
##  $ cum_death_August       : int  NA NA NA NA NA NA 165 NA 67 NA ...
##  $ cum_death_September    : int  NA NA NA NA NA NA 465 NA 155 NA ...
##  $ cum_death_October      : int  NA NA NA NA NA NA 538 NA 180 NA ...
##  $ cum_death_November     : int  1 NA NA NA NA NA 593 NA 186 NA ...
##  $ cum_death_December     : int  16 NA 3 NA NA NA 613 NA 190 3 ...
##  - attr(*, ".internal.selfref")=<externalptr>

##  observation_id  firstcase_date_id firstcase_continent_id   country
##  Min.   :  1.0   Min.   :  1.00    Min.   : 11.0          Length:179
##  1st Qu.: 45.5   1st Qu.: 44.25    1st Qu.:133.2          Class :character
##  Median : 90.0   Median : 87.50    Median :245.5          Mode  :character
##  Mean   : 90.0   Mean   : 87.71    Mean   :288.4
##  3rd Qu.:134.5   3rd Qu.:130.75    3rd Qu.:403.8
##  Max.   :179.0   Max.   :175.00    Max.   :621.0
##                  NA's   :5         NA's   :5
##  firstcasereport_date cum_case_April   cum_case_May    cum_case_June
##  Length:179           Min.   : 7.00   Min.   :  1.00   Min.   :   1.0
##  Class :character     1st Qu.: 9.75   1st Qu.:  1.00   1st Qu.:   2.0
##  Mode  :character     Median :12.50   Median :  4.00   Median :   5.0
##                       Mean   :12.50   Mean   : 28.23   Mean   : 276.4
##                       3rd Qu.:15.25   3rd Qu.: 13.00   3rd Qu.:  28.5
##                       Max.   :18.00   Max.   :156.00   Max.   :8975.0
##                       NA's   :177     NA's   :166      NA's   :116
##  cum_case_July     cum_case_August   cum_case_Aug09  firstdeath_date_id
##  Min.   :    1.0   Min.   :    1.0   Min.   :    1   Min.   :  1.00
##  1st Qu.:    4.0   1st Qu.:    6.0   1st Qu.:    9   1st Qu.: 31.50
##  Median :   18.5   Median :   51.0   Median :   55   Median : 62.00
##  Mean   :  659.4   Mean   : 1140.3   Mean   : 1206   Mean   : 61.97
##  3rd Qu.:  154.2   3rd Qu.:  475.5   3rd Qu.:  507   3rd Qu.: 92.50
##  Max.   :27717.0   Max.   :43771.0   Max.   :43771   Max.   :123.00
##  NA's   :61        NA's   :12        NA's   :5       NA's   :56
##  firstdeath_continent_id firstdeath_date    cum_death_May cum_death_June
##  Min.   : 11.0           Length:179         Min.   :1     Min.   : 1.00
##  1st Qu.:206.5           Class :character   1st Qu.:3     1st Qu.: 1.75
##  Median :329.0           Mode  :character   Median :5     Median : 8.50
##  Mean   :347.8                              Mean   :5     Mean   :28.75
##  3rd Qu.:514.5                              3rd Qu.:7     3rd Qu.:35.50
##  Max.   :613.0                              Max.   :9     Max.   :97.00
##  NA's   :56                                 NA's   :177   NA's   :175
##  cum_death_July   cum_death_August cum_death_September cum_death_October
##  Min.   :  1.00   Min.   :  1.00   Min.   :  1.00      Min.   :  1.00
##  1st Qu.:  1.00   1st Qu.:  1.00   1st Qu.:  1.00      1st Qu.:  2.00
##  Median :  2.00   Median :  6.50   Median :  8.00      Median :  8.00
##  Mean   : 19.53   Mean   : 27.67   Mean   : 44.97      Mean   : 53.48
##  3rd Qu.: 12.00   3rd Qu.: 21.25   3rd Qu.: 27.00      3rd Qu.: 31.50
##  Max.   :127.00   Max.   :353.00   Max.   :557.00      Max.   :899.00
##  NA's   :162      NA's   :133      NA's   :110         NA's   :96
##  cum_death_November cum_death_December
##  Min.   :   1.00    Min.   :   1.00
##  1st Qu.:   2.00    1st Qu.:   3.00
```

```
##  Median :   7.00    Median :  12.00
##  Mean   :  62.17    Mean   :  71.12
##  3rd Qu.:  34.50    3rd Qu.:  40.50
##  Max.   :1368.00    Max.   :1528.00
##  NA's   :80         NA's   :56
```

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).

```
SwineFlu$firstcasereport_date <- as.Date(SwineFlu$firstcasereport_date,format
= "%m/%d/%Y")
SwineFlu$firstdeath_date <- as.Date(SwineFlu$firstdeath_date,format =
"%m/%d/%Y")
if(demo) {str(SwineFlu)}

## Classes 'data.table' and 'data.frame':   179 obs. of  22 variables:
##  $ observation_id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ firstcase_date_id      : int  137 154 99 161 170 101 28 122 29 9 ...
##  $ firstcase_continent_id : int  335 243 604 244 135 122 503 124 402 203
...
##  $ country                : chr  "Afghanistan" "Albania" "Algeria"
"Andorra" ...
##  $ firstcasereport_date   : Date, format: "2009-07-08" "2009-07-22" ...
##  $ cum_case_April         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_case_May           : int  NA NA NA NA NA NA NA NA NA 1 ...
##  $ cum_case_June          : int  NA NA NA NA NA NA 100 NA 297 1 ...
##  $ cum_case_July          : int  NA NA 2 NA NA 2 1587 NA 4090 15 ...
##  $ cum_case_August        : int  32 3 16 1 NA 3 3056 13 22109 153 ...
##  $ cum_case_Aug09         : int  32 13 19 1 1 4 5710 13 24949 192 ...
##  $ firstdeath_date_id     : int  99 NA 123 NA NA NA 9 NA 11 100 ...
##  $ firstdeath_continent_id: int  533 NA 613 NA NA NA 203 NA 401 324 ...
##  $ firstdeath_date        : Date, format: "2009-10-30" NA ...
##  $ cum_death_May          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_June         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_July         : int  NA NA NA NA NA NA 26 NA 7 NA ...
##  $ cum_death_August       : int  NA NA NA NA NA NA 165 NA 67 NA ...
##  $ cum_death_September    : int  NA NA NA NA NA NA 465 NA 155 NA ...
##  $ cum_death_October      : int  NA NA NA NA NA NA 538 NA 180 NA ...
##  $ cum_death_November     : int  1 NA NA NA NA NA 593 NA 186 NA ...
##  $ cum_death_December     : int  16 NA 3 NA NA NA 613 NA 190 3 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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

```
SwineFlu$Datediff_calc <-
difftime(SwineFlu$firstcasereport_date,as.Date("2009-04-29"),units = "days")
if(demo) {str(SwineFlu)}

## Classes 'data.table' and 'data.frame':   179 obs. of  23 variables:
##  $ observation_id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ firstcase_date_id      : int  137 154 99 161 170 101 28 122 29 9 ...
```

```
##  $ firstcase_continent_id : int  335 243 604 244 135 122 503 124 402 203
...
##  $ country                : chr  "Afghanistan" "Albania" "Algeria"
"Andorra" ...
##  $ firstcasereport_date   : Date, format: "2009-07-08" "2009-07-22" ...
##  $ cum_case_April         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_case_May           : int  NA NA NA NA NA NA NA NA NA 1 ...
##  $ cum_case_June          : int  NA NA NA NA NA NA 100 NA 297 1 ...
##  $ cum_case_July          : int  NA NA 2 NA NA 2 1587 NA 4090 15 ...
##  $ cum_case_August        : int  32 3 16 1 NA 3 3056 13 22109 153 ...
##  $ cum_case_Aug09         : int  32 13 19 1 1 4 5710 13 24949 192 ...
##  $ firstdeath_date_id     : int  99 NA 123 NA NA NA 9 NA 11 100 ...
##  $ firstdeath_continent_id: int  533 NA 613 NA NA NA 203 NA 401 324 ...
##  $ firstdeath_date        : Date, format: "2009-10-30" NA ...
##  $ cum_death_May          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_June         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cum_death_July         : int  NA NA NA NA NA NA 26 NA 7 NA ...
##  $ cum_death_August       : int  NA NA NA NA NA NA 165 NA 67 NA ...
##  $ cum_death_September    : int  NA NA NA NA NA NA 465 NA 155 NA ...
##  $ cum_death_October      : int  NA NA NA NA NA NA 538 NA 180 NA ...
##  $ cum_death_November     : int  1 NA NA NA NA NA 593 NA 186 NA ...
##  $ cum_death_December     : int  16 NA 3 NA NA NA 613 NA 190 3 ...
##  $ Datediff_calc          : 'difftime' num  70 84 54 61 ...
##   ..- attr(*, "units")= chr "days"
##  - attr(*, ".internal.selfref")=<externalptr>
```

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)

```
SwineFlu2009_days_from_first_incidence <- subset(SwineFlu,select =
c("firstcase_date_id","country","Datediff_calc"))

fwrite(SwineFlu2009_days_from_first_incidence,"SwineFlu2009_days_from_first_i
ncidence.csv")
str(SwineFlu2009_days_from_first_incidence)
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  3 variables:
##  $ firstcase_date_id: int  137 154 99 161 170 101 28 122 29 9 ...
##  $ country          : chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
##  $ Datediff_calc    : 'difftime' num  70 84 54 61 ...
##   ..- attr(*, "units")= chr "days"
##  - attr(*, ".internal.selfref")=<externalptr>
```

Section -2

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

```
pizza <- fread("Pizza.csv")

str(pizza)
```

```
## Classes 'data.table' and 'data.frame':   120 obs. of  6 variables:
##  $ SurveyNum: int  101 102 103 104 105 106 107 108 109 110 ...
##  $ Arugula  : int  1 5 4 5 3 2 2 1 4 2 ...
##  $ PineNut  : int  3 4 2 3 5 3 5 3 3 2 ...
##  $ Squash   : int  3 2 5 2 5 1 5 1 3 5 ...
##  $ Shrimp   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Eggplant : int  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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

```
print(pizza)
```

```
##      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      5      1       NA
## 119:      1209      NA       4      5      3       NA
## 120:      1210      NA       5      5      1       NA
```

c. Examine the class of each column of data.

```
lapply(pizza,class)
```

```
## $SurveyNum
## [1] "integer"
##
## $Arugula
## [1] "integer"
##
## $PineNut
## [1] "integer"
##
## $Squash
## [1] "integer"
##
## $Shrimp
## [1] "integer"
##
## $Eggplant
## [1] "integer"
```

d. Print the summary statistics of the data using describe() in “psych” package.

```
describe(pizza)
```

```
##           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  0.13
## Squash       -1.43  0.17
## Shrimp       -1.20  0.14
## Eggplant     -1.57  0.21
```

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.

Ans: Survey Number columns in the data frame should be a factor variable as it is unique and is not ordinal in nature. The fread() function identifies the column as an integer and assume it's an continuous variable.

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

```
pizza <- fread("Pizza.csv",header = T,colClasses =
c("factor","integer","integer","integer","integer","integer"))
if(demo) {str(pizza)}

## Classes 'data.table' and 'data.frame':   120 obs. of  6 variables:
##  $ SurveyNum: Factor w/ 120 levels "0101","0102",..: 1 2 3 4 5 6 7 8 9 10
...
##  $ Arugula  : int  1 5 4 5 3 2 2 1 4 2 ...
##  $ PineNut  : int  3 4 2 3 5 3 5 3 3 2 ...
##  $ Squash   : int  3 2 5 2 5 1 5 1 3 5 ...
##  $ Shrimp   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Eggplant : int  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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().)

```
pizza$avg_rating <- rowMeans(pizza[,2:6],na.rm = T)
if(demo) {str(pizza)
  describe(pizza)
  }
```

```
## Classes 'data.table' and 'data.frame':   120 obs. of  7 variables:
##  $ SurveyNum : Factor w/ 120 levels "0101","0102",..: 1 2 3 4 5 6 7 8 9 10
...
##  $ Arugula   : int  1 5 4 5 3 2 2 1 4 2 ...
##  $ PineNut   : int  3 4 2 3 5 3 5 3 3 2 ...
##  $ Squash    : int  3 2 5 2 5 1 5 1 3 5 ...
##  $ Shrimp    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Eggplant  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_rating: num  2.33 3.67 3.67 3.33 4.33 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##            vars   n  mean    sd median trimmed   mad min    max  range
skew
## SurveyNum*    1 120 60.50 34.79   60.5   60.50 44.48   1 120.00 119.00
0.00
## Arugula       2  40  3.08  1.49    3.0    3.09  1.48   1   5.00   4.00 -
0.12
## PineNut       3 100  3.14  1.29    3.0    3.17  1.48   1   5.00   4.00
0.02
## Squash        4  80  3.16  1.51    3.0    3.20  2.22   1   5.00   4.00 -
0.14
## Shrimp        5  90  2.97  1.33    3.0    2.96  1.48   1   5.00   4.00 -
0.03
## Eggplant      6  50  2.86  1.51    3.0    2.83  1.48   1   5.00   4.00 -
0.01
## avg_rating    7 120  3.06  0.76    3.0    3.06  0.99   1   4.67   3.67 -
0.16
##            kurtosis   se
## SurveyNum*    -1.23 3.18
## 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
## avg_rating    -0.61 0.07
```

Section - 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.

Ans: The fread() function creates a data frame using the first row of the data its reading. So, the initial data frame which was created has 11 columns in it. But the data has 12 columns in it. The data has an additional column when the internet_usage flag is YES. To solve this problem, I have used read.csv() function and manipulated the df accordingly.

```
hotel <- fread("Hotel.csv")

## Warning in fread("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>>

hotel <- fread("Hotel.csv", fill=TRUE, na.strings = c("NA", ""),
               sep = "auto", data.table = TRUE, stringsAsFactors = FALSE)
if(demo) {str(hotel)}

## Classes 'data.table' and 'data.frame':   179 obs. of  12 variables:
##  $ V1 : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ V2 : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3 : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4 : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5 : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
##  $ V6 : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7 : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8 : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
##  $ V9 : chr  "NO" "NO" "NO" "YES" ...
##  $ V10: chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11: chr  "295" "75" "255" "Basic w/view" ...
##  $ V12: int  NA NA NA 155 NA NA 155 195 295 75 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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.

```
colnames(hotel)[1] <- "room_no"
colnames(hotel)[2] <- "no_of_guests"
if(demo) {str(hotel)}

## Classes 'data.table' and 'data.frame':   179 obs. of  12 variables:
##  $ room_no     : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests: int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3          : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4          : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5          : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V6          : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7          : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8          : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V9          : chr  "NO" "NO" "NO" "YES" ...
##  $ V10         : chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11         : chr  "295" "75" "255" "Basic w/view" ...
##  $ V12         : int  NA NA NA 155 NA NA 155 195 295 75 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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

```
hotel$check_in_date <- as.Date(with(hotel,paste(V3,V4,V5,sep="-")),"%m-%d-
%Y")
hotel$check_out_date <- as.Date(with(hotel,paste(V6,V7,V8,sep="-")),"%m-%d-
%Y")
if(demo) {str(hotel)}
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  14 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3            : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4            : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V6            : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7            : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V9            : chr  "NO" "NO" "NO" "YES" ...
##  $ V10           : chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11           : chr  "295" "75" "255" "Basic w/view" ...
##  $ V12           : int  NA NA NA 155 NA NA 155 195 295 75 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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())

```
hotel[,':='(internet_usage=as.numeric(ifelse(V9 == "YES",V10,0)))]
if(demo) {str(hotel)}
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  15 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3            : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4            : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V6            : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7            : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V9            : chr  "NO" "NO" "NO" "YES" ...
##  $ V10           : chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11           : chr  "295" "75" "255" "Basic w/view" ...
##  $ V12           : int  NA NA NA 155 NA NA 155 195 295 75 ...
```

```
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

e. Using the data.table syntax, create a column of room type.

```
hotel[,':='(room_type=ifelse(V9 ==
"YES",as.character(V11),as.character(V10)))]
if(demo) {str(hotel)}
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  16 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3            : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4            : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V6            : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7            : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V9            : chr  "NO" "NO" "NO" "YES" ...
##  $ V10           : chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11           : chr  "295" "75" "255" "Basic w/view" ...
##  $ V12           : int  NA NA NA 155 NA NA 155 195 295 75 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
##  $ room_type     : chr  "Deluxe Suite" "Basic no view" "Suite" "Basic
w/view" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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)

```
hotel[,':='(room_rate=ifelse(V9 == "YES",V12,strtoi(V11)))]
if(demo) {str(hotel)}
```

```
## Classes 'data.table' and 'data.frame':   179 obs. of  17 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ V3            : int  2 2 2 2 2 2 1 2 2 2 ...
##  $ V4            : int  7 2 2 3 3 7 31 1 3 7 ...
##  $ V5            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V6            : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V7            : int  11 12 13 12 12 13 13 13 12 11 ...
##  $ V8            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014
...
##  $ V9            : chr  "NO" "NO" "NO" "YES" ...
```

```
##  $ V10           : chr  "Deluxe Suite" "Basic no view" "Suite" "2" ...
##  $ V11           : chr  "295" "75" "255" "Basic w/view" ...
##  $ V12           : int  NA NA NA 155 NA NA 155 195 295 75 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
##  $ room_type     : chr  "Deluxe Suite" "Basic no view" "Suite" "Basic
w/view" ...
##  $ room_rate     : int  295 75 255 155 195 255 155 195 295 75 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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.

```
hotelColNames <-
c("room_no","no_of_guests","check_in_date","check_out_date","internet_usage",
"room_type","room_rate")
hotel_cleaned <- subset(hotel,select = hotelColNames)
if(demo) {str(hotel_cleaned)}

## Classes 'data.table' and 'data.frame':   179 obs. of  7 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
##  $ room_type     : chr  "Deluxe Suite" "Basic no view" "Suite" "Basic
w/view" ...
##  $ room_rate     : int  295 75 255 155 195 255 155 195 295 75 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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).

```
hotel_cleaned$subtotal <- hotel_cleaned$room_rate *
as.numeric(difftime(hotel_cleaned$check_out_date,hotel_cleaned$check_in_date,
units = "days")) + 10 * (hotel_cleaned$no_of_guests-1) *
as.numeric(difftime(hotel_cleaned$check_out_date,
hotel_cleaned$check_in_date, units="days")) +
ifelse(hotel_cleaned$internet_usage > 0 , 9.95 + 5.95 *
(hotel_cleaned$internet_usage),0)
if(demo) {str(hotel_cleaned)}

## Classes 'data.table' and 'data.frame':   179 obs. of  8 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
```

```
##  $ room_type     : chr  "Deluxe Suite" "Basic no view" "Suite" "Basic
w/view" ...
##  $ room_rate     : int  295 75 255 155 195 255 155 195 295 75 ...
##  $ subtotal      : num  1260 850 3135 1777 1935 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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.

```
hotel_cleaned$total <- round(hotel_cleaned$subtotal * 1.0875,2)
if(demo) {str(hotel_cleaned)}

## Classes 'data.table' and 'data.frame':   179 obs. of  9 variables:
##  $ room_no       : int  211 214 216 220 221 223 238 241 244 247 ...
##  $ no_of_guests  : int  3 2 4 5 3 5 4 1 5 4 ...
##  $ check_in_date : Date, format: "2014-02-07" "2014-02-02" ...
##  $ check_out_date: Date, format: "2014-02-11" "2014-02-12" ...
##  $ internet_usage: num  0 0 0 2 0 0 10 3 9 4 ...
##  $ room_type     : chr  "Deluxe Suite" "Basic no view" "Suite" "Basic
w/view" ...
##  $ room_rate     : int  295 75 255 155 195 255 155 195 295 75 ...
##  $ subtotal      : num  1260 850 3135 1777 1935 ...
##  $ total         : num  1370 924 3409 1932 2104 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

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.

```
(results <- hotel_cleaned[which(hotel_cleaned$room_no == 247 &
hotel_cleaned$check_in_date == "2014-02-07"),])

##    room_no no_of_guests check_in_date check_out_date internet_usage
## 1:     247            4    2014-02-07     2014-02-11              4
##        room_type room_rate subtotal  total
## 1: Basic no view        75   453.75 493.45

results$total

## [1] 493.45
```