RWorksheet_calvario#3b.Rmd

Jolien

2024-10-02

1. Create a data frame using the table below.

a. Write the codes.

```
respondents <- c(1:20)
respondents
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
sex <- c(2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2)
sex
```

```
##  [1] 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2
```

```
occupation <- c(1, 2, 2, 2, 1, 2, 3, 1, 1, 1, 3, 2, 1, 3, 3, 1, 3, 1, 2, 1)
occupation
```

```
##  [1] 1 2 2 2 1 2 3 1 1 1 3 2 1 3 3 1 3 1 2 1
```

```
persons <- c(5, 7, 3, 8, 5, 9, 6, 7, 8, 4, 7, 5, 4, 7, 8, 8, 3, 11, 7, 6)
persons
```

```
##  [1]  5  7  3  8  5  9  6  7  8  4  7  5  4  7  8  8  3 11  7  6
```

```
siblings <- c(6, 4, 4, 1, 2, 1, 5, 3, 1, 2, 3, 2, 5, 5, 2, 1, 2, 5, 3, 2)
siblings
```

```
##  [1] 6 4 4 1 2 1 5 3 1 2 3 2 5 5 2 1 2 5 3 2
```

```
houses <- c(1, 2, 3, 1, 1, 3, 3, 1, 2, 3, 2, 3, 2, 2, 3, 3, 3, 3, 3, 2)
houses
```

```
##  [1] 1 2 3 1 1 3 3 1 2 3 2 3 2 2 3 3 3 3 3 2
```

```
data <- data.frame(respondents, sex, occupation, persons, siblings, houses)
data
```

```
##    respondents sex occupation persons siblings houses
## 1            1   2          1       5        6      1
## 2            2   2          2       7        4      2
## 3            3   1          2       3        4      3
## 4            4   2          2       8        1      1
## 5            5   2          1       5        2      1
## 6            6   2          2       9        1      3
## 7            7   2          3       6        5      3
## 8            8   2          1       7        3      1
```

```
## 9            9   2          1       8        1      2
## 10          10   2          1       4        2      3
## 11          11   1          3       7        3      2
## 12          12   2          2       5        2      3
## 13          13   2          1       4        5      2
## 14          14   2          3       7        5      2
## 15          15   2          3       8        2      3
## 16          16   2          1       8        1      3
## 17          17   2          3       3        2      3
## 18          18   2          1      11        5      3
## 19          19   1          2       7        3      3
## 20          20   2          1       6        2      2
```

1. b. Describe the data. Get the structure or the summary of the data

```
str(data)
```

```
## 'data.frame':    20 obs. of  6 variables:
##  $ respondents: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ sex        : num  2 2 1 2 2 2 2 2 2 2 ...
##  $ occupation : num  1 2 2 2 1 2 3 1 1 1 ...
##  $ persons    : num  5 7 3 8 5 9 6 7 8 4 ...
##  $ siblings   : num  6 4 4 1 2 1 5 3 1 2 ...
##  $ houses     : num  1 2 3 1 1 3 3 1 2 3 ...
```

```
summary(data)
```

```
##   respondents         sex         occupation      persons        siblings
##  Min.   : 1.00   Min.   :1.00   Min.   :1.00   Min.   : 3.0   Min.   :1.00
##  1st Qu.: 5.75   1st Qu.:2.00   1st Qu.:1.00   1st Qu.: 5.0   1st Qu.:2.00
##  Median :10.50   Median :2.00   Median :2.00   Median : 7.0   Median :2.50
##  Mean   :10.50   Mean   :1.85   Mean   :1.80   Mean   : 6.4   Mean   :2.95
##  3rd Qu.:15.25   3rd Qu.:2.00   3rd Qu.:2.25   3rd Qu.: 8.0   3rd Qu.:4.25
##  Max.   :20.00   Max.   :2.00   Max.   :3.00   Max.   :11.0   Max.   :6.00
##      houses
##  Min.   :1.0
##  1st Qu.:2.0
##  Median :2.5
##  Mean   :2.3
##  3rd Qu.:3.0
##  Max.   :3.0
```

c. Is the mean number of siblings attending is 5?

No

```
mean_num <- mean(siblings)
mean_num
```

```
## [1] 2.95
```

d. Extract the 1st two rows and then all the columns using the subsetting functions. Write the codes and its output.

```
extract <- data[1:2, ]
extract
```

```
##   respondents sex occupation persons siblings houses
## 1           1   2          1       5        6      1
## 2           2   2          2       7        4      2
```

e. Extract 3rd and 5th row with 2nd and 4th column. Write the codes and its result.

```
extracted_data <- data[c(3, 5), c(2, 4)]
(extracted_data)
```

```
##   sex persons
## 3   1       3
## 5   2       5
```

f. Select the variable types of houses then store the vector that results as types_houses. Write the codes.

```
types_houses <- class(houses)
types_houses
```

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

g. Select only all Males respondent that their father occupation was farmer. Write the codes and its output.

```
male_farmers <- data[data$Sex == 1 & data$Occupation == 1, ]
print(male_farmers)
```

```
## [1] respondents sex         occupation  persons     siblings    houses
## <0 rows> (or 0-length row.names)
```

h. Select only all females respondent that have greater than or equal to 5 number of siblings attending school. Write the codes and its outputs.

```
female <-  data[data$Sex == 2 & data$Siblings >= 5, ]
print(female)
```

```
## [1] respondents sex         occupation  persons     siblings    houses
## <0 rows> (or 0-length row.names)
```

2. Write a R program to create an empty data frame. Using the following codes:

```
df = data.frame(Ints=integer(),

Doubles=double(), Characters=character(),
Logicals=logical(),
Factors=factor(),
stringsAsFactors=FALSE)
print("Structure of the empty dataframe:")
```

```
## [1] "Structure of the empty dataframe:"
```

```
str(df)
```

```
## 'data.frame':    0 obs. of  5 variables:
##  $ Ints      : int
##  $ Doubles   : num
##  $ Characters: chr
##  $ Logicals  : logi
##  $ Factors   : Factor w/ 0 levels:
```

a. Describe the results.

The provided R code creates an empty data frame named df with five specified column types: integers, doubles, characters, logicals, and factors. When the structure of this empty data frame is printed using the str() function, it reveals that each column labeled Ints, Doubles, Characters, Logicals, and Factors is defined with the appropriate data type, but all columns contain zero observations (rows). The argument stringsAsFactors = FALSE ensures that any character data remains as character type rather than being automatically converted to factors. Overall, the output confirms that while the data frame is correctly structured to hold various data types, it currently holds no data.

3. Create a .csv file of this. Save it as HouseholdData.csv

```
household_data <- data.frame(
  Respondents = 1:10,
  Sex = c("Male", "Female", "Female", "Male", "Male", "Female", "Female", "Male", "Female", "Male"),
  Fathers_Occupation = c(1, 2, 3, 3, 1, 2, 2, 1, 1, 3),
  Persons_at_Home = c(5, 7, 3, 8, 6, 4, 2, 4, 11, 6),
  Siblings_at_School = c(5, 3, 3, 5, 6, 3, 1, 2, 6, 6),
  Types_of_Houses = c("Wood", "Concrete", "Concrete", "Wood", "Semi-concrete", "Semi-concrete", "Wood",
household_data
```

```
##    Respondents    Sex Fathers_Occupation Persons_at_Home Siblings_at_School
```

```
## 1            1   Male                 1              5                 5
## 2            2 Female                 2              7                 3
## 3            3 Female                 3              3                 3
## 4            4   Male                 3              8                 5
## 5            5   Male                 1              6                 6
## 6            6 Female                 2              4                 3
## 7            7 Female                 2              2                 1
## 8            8   Male                 1              4                 2
## 9            9 Female                 1             11                 6
## 10          10   Male                 3              6                 6
##    Types_of_Houses
## 1             Wood
## 2         Concrete
## 3         Concrete
## 4             Wood
## 5    Semi-concrete
## 6    Semi-concrete
## 7             Wood
## 8    Semi-concrete
## 9    Semi-concrete
## 10        Concrete
```

```
write.csv(df, "HouseholdData.csv", row.names = FALSE)
```

a. Import the csv file into the R environment. Write the codes.

```
household_data <- read.csv("HouseholdData.csv")

print(household_data)
```

```
## [1] Ints       Doubles    Characters Logicals   Factors
## <0 rows> (or 0-length row.names)
```

b. Convert the Sex into factor using factor() function and change it into integer.[Legend: Male = 1 and Female = 2]. Write the R codes and its output.

```
household_data$Sex <- factor(household_data$Sex, levels = c("Male", "Female"), labels = c(1, 2))

household_data$Sex <- as.integer(household_data$Sex)
print(household_data)
```

```
## [1] Ints       Doubles    Characters Logicals   Factors    Sex
## <0 rows> (or 0-length row.names)
```

c. Convert the Type of Houses into factor and change it into integer. [Legend: Wood = 1; Congrete = 2; Semi-Congrete = 3]. Write the R codes and its output.

```
household_data$Types_of_Houses <- factor(household_data$Types_of_Houses, levels = c("Wood", "Concrete",

household_data$Types_of_Houses <- as.integer(household_data$Types_of_Houses)

print(household_data)
```

```
## [1] Ints            Doubles         Characters      Logicals
## [5] Factors         Sex             Types_of_Houses
## <0 rows> (or 0-length row.names)
```

d. On father's occupation, factor it as Farmer = 1; Driver = 2; and Others = 3. What is the R code and its output?

```
household_data$Fathers_Occupation <- factor(household_data$Fathers_Occupation, levels = c(1, 2, 3), labe

household_data$Fathers_Occupation <- as.integer(household_data$Fathers_Occupation)

print(household_data)
```

```
## [1] Ints              Doubles           Characters        Logicals
## [5] Factors           Sex               Types_of_Houses   Fathers_Occupation
## <0 rows> (or 0-length row.names)
```

e. Select only all females respondent that has a father whose occupation is driver. Write the codes and its output.

```
fem_driver_respondents <- subset(household_data, Sex == 2 & Fathers_Occupation == 2)
print(fem_driver_respondents)
```

```
## [1] Ints              Doubles           Characters        Logicals
## [5] Factors           Sex               Types_of_Houses   Fathers_Occupation
## <0 rows> (or 0-length row.names)
```

f. Select the respondents that have greater than or equal to 5 number of siblings attending school. Write the codes and its output.

```
siblings_gte_5 <- subset(household_data, siblings >= 5)
print(siblings_gte_5)
```

```
##      Ints Doubles Characters Logicals Factors Sex Types_of_Houses
## NA     NA      NA         NA       NA      NA  NA              NA
## NA.1   NA      NA         NA       NA      NA  NA              NA
## NA.2   NA      NA         NA       NA      NA  NA              NA
## NA.3   NA      NA         NA       NA      NA  NA              NA
## NA.4   NA      NA         NA       NA      NA  NA              NA
```

```
##      Fathers_Occupation
## NA                   NA
## NA.1                 NA
## NA.2                 NA
## NA.3                 NA
## NA.4                 NA
```

4. Interpret the graph.

The bar graph titled "Sentiments of Tweets Per Day" provides a visual representation of tweet sentiments over the week from July 14 to July 21, 2020. The sentiments are classified into three categories: negative (red), neutral (orange), and positive (blue).

The data indicates that negative sentiments were the most prevalent throughout the week, peaking on July 15 and July 20, when they surpassed 4,000 tweets. This suggests an increase in negative discussions or reactions on social media during these days. Neutral sentiments are visible but consistently lower than negative ones, reflecting a more reserved tone in some tweets. Conversely, positive sentiments were the least frequent, indicating that optimistic responses were uncommon during this period.

In conclusion, the graph depicts a timeframe marked by a significant presence of negativity in social media interactions, with a notable scarcity of positive or neutral sentiments. This insight could be valuable for understanding public reactions to particular events or trending topics during this timeframe.