RWorkheet_nandin#4a.Rmd

Mary coolen nandin

2024-10-19

1. The table below shows the data about shoe size and height. Create a data frame.

```
table <- data.frame(
Shoe_size = c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5,
10.5, 8.5, 12.0, 10.5, 13.0, 11.5, 8.5, 5.0, 10.0, 6.5,7.5, 8.5, 10.5, 8.5, 10.5, 11.0, 9.0, 13.0),
Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0,
74.5, 67.0, 71.0, 71.0, 77.0, 72.0, 59.0, 62.0, 72.0, 66.0,64.0, 67.0, 73.0, 69.0, 72.0, 70.0, 69.0, 70
Gender = c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F",
"M", "F", "M", "M", "M", "M", "F", "F", "M", "F","F", "M", "M", "F", "M", "M", "M", "M")
)
table
```

```
##    Shoe_size Height Gender
## 1        6.5   66.0      F
## 2        9.0   68.0      F
## 3        8.5   64.5      F
## 4        8.5   65.0      F
## 5       10.5   70.0      M
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

a. Describe the data. The data frame Table contains three columns: Shoe_size, Height, and Gender. Shoe_size: A numeric vector representing the shoe sizes of individuals. The values range from 5.0 to

13.0. Height: A numeric vector representing the height of individuals in inches. The values range from 59.0 to 77.0 1 inches. Gender: A categorical variable indicating the gender of each individual, with values “M” for male and “F” for female.

b. Create a subset by males and females with their corresponding shoe size and height. What its result? Show the R scripts.

```
males <- subset(table, Gender == "M", select = c(Shoe_size, Height))
females <- subset(table, Gender == "F", select = c(Shoe_size, Height))
males
```

```
##    Shoe_size Height
## 5       10.5   70.0
## 9       13.0   72.0
## 11      10.5   74.5
## 13      12.0   71.0
## 14      10.5   71.0
## 15      13.0   77.0
## 16      11.5   72.0
## 19      10.0   72.0
## 22       8.5   67.0
## 23      10.5   73.0
## 25      10.5   72.0
## 26      11.0   70.0
## 27       9.0   69.0
## 28      13.0   70.0
```

```
females
```

```
##    Shoe_size Height
## 1        6.5   66.0
## 2        9.0   68.0
## 3        8.5   64.5
## 4        8.5   65.0
## 6        7.0   64.0
## 7        9.5   70.0
## 8        9.0   71.0
## 10       7.5   64.0
## 12       8.5   67.0
## 17       8.5   59.0
## 18       5.0   62.0
## 20       6.5   66.0
## 21       7.5   64.0
## 24       8.5   69.0
```

c. Find the mean of shoe size and height of the respondents. Write the R scripts and its result.

```
MeanShoeSize <- mean(table$Shoe_size)
MeanShoeSize
```

```
## [1] 9.410714
```

```
meanHeight <- mean (table$Height)
meanHeight
```

```
## [1] 68.57143
```

d. Is there a relationship between shoe size and height? Why?

Yes, there is a relationship between the shoe size and height of the respondents for the reason that thethe shoesize aligns to their height.

Figure 1: Household Data

Factors

A nominal variable is a categorical variable without an implied order. This means that it is impossible to say that 'one is worth more than the other'. In contrast, ordinal variables do have a natural ordering.

Example:

```
Gender <- c("M","F","F","M")
factor_Gender <- factor(Gender)
factor_Gender
```

```
## [1] M F F M
## Levels: F M
```

2. Construct character vector months to a factor with factor() and assign the result to factor_months_vector. Print out factor_months_vector and assert that R prints out the factor levels below the actual values.

Consider data consisting of the names of months: "March","April","January","November","January", "September","October","September","November","August", "January","November","November","February","May","August",

"July","December","August","August","September","November","February",| April")

```
months_vector <- c("March","April","January","November","January","September","October","September","No
factor_months_vector <- factor (months_vector)
factor_months_vector
```

```
##  [1] March     April     January   November  January   September October
##  [8] September November  January   November  November  February  May
## [15] August    July      December  August    August    September April
## 11 Levels: April August December February January July March May ... September
```

3. Then check the summary() of the months_vector and factor_months_vector. | Inter- pret the results of both vectors. Are they both equally useful in this case?

```
summary(months_vector)
```

```
##    Length     Class      Mode
##        21 character character
```

```
summary(factor_months_vector)
```

```
##     April    August  December  February   January      July     March       May
##         2         3         1         1         3         1         1         1
##  November   October September
##         4         1         3
```

4. Create a vector and factor for the table below.

```
factor_data <-c("East","West","North")
levels <- c(1, 4, 3)
```

```
levels
```

```
## [1] 1 4 3
```

```
new_order_data <- factor(factor_data,levels = c("East","West","North"))
print(new_order_data)
```

```
## [1] East  West  North
## Levels: East West North
```

5. Enter the data below in Excel with file name = import_march.csv

```
library(readr)
import_march <- read_csv("/cloud/project/import_march.csv")
```

```
## Rows: 6 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Students
## dbl (3): Strategy 1, Strategy 2, Strategy 3
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
import_march
```

```
## # A tibble: 6 x 4
##   Students `Strategy 1` `Strategy 2` `Strategy 3`
##   <chr>           <dbl>        <dbl>        <dbl>
## 1 Male                8           10            8
## 2 <NA>                4            8            6
## 3 <NA>                0            6            4
## 4 Female             14            4           15
## 5 <NA>               10            2           12
## 6 <NA>                6            0            9
```