RWorksheet_6.Rmd

Kent Hervey D. Gener BSIT2-B

2023-12-13

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

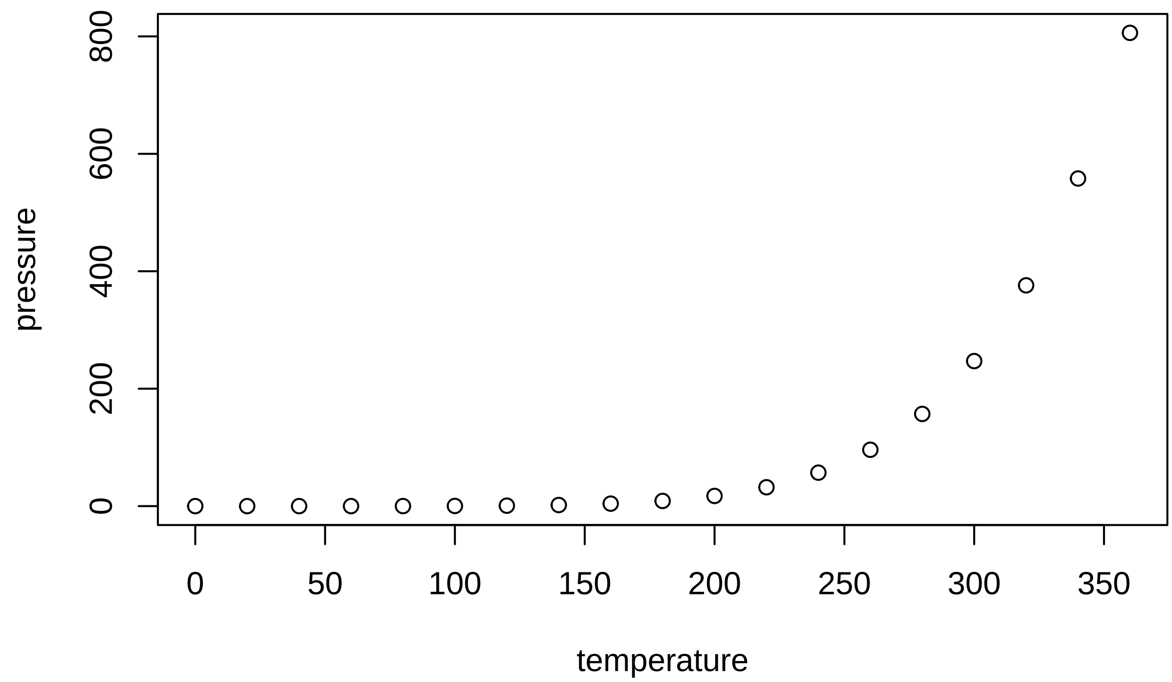
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Basic Statistics 1. Create a data frame for the table below. Show your solution.

```
df <- data.frame(
  Student = c(1, 2, 2, 4, 5, 7, 8, 9, 10),
  Pre_test = c(55, 54, 47, 57, 51, 61, 57, 54, 63),
  Post_test = c(61, 60, 56, 63, 56, 63, 59, 56, 62)
)
print(df)
```

```
##   Student Pre_test Post_test
## 1       1       55        61
## 2       2       54        60
## 3       2       47        56
## 4       4       57        63
## 5       5       51        56
## 6       7       61        63
## 7       8       57        59
## 8       9       54        56
## 9      10       63        62
```

a. Compute the descriptive statistics using different packages (Hmisc and pastecs). Write the codes and its result.

```r
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.3.2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(pastecs)
```

```
## Warning: package 'pastecs' was built under R version 4.3.2
```

```r
desc_stats_hmisc <- Hmisc::describe(df)
print(desc_stats_hmisc)
```

```
## df
##
##  3  Variables      9  Observations
## --------------------------------------------------------------------------------
## Student
##        n  missing distinct     Info     Mean      Gmd
##        9        0        8    0.992    5.333        4
##
## Value          1     2     4     5     7     8     9    10
## Frequency      1     2     1     1     1     1     1     1
## Proportion 0.111 0.222 0.111 0.111 0.111 0.111 0.111 0.111
##
## For the frequency table, variable is rounded to the nearest 0
## --------------------------------------------------------------------------------
## Pre_test
##        n  missing distinct     Info     Mean      Gmd
##        9        0        7    0.983    55.44    5.722
##
## Value         47    51    54    55    57    61    63
## Frequency      1     1     2     1     2     1     1
## Proportion 0.111 0.111 0.222 0.111 0.222 0.111 0.111
##
## For the frequency table, variable is rounded to the nearest 0
## --------------------------------------------------------------------------------
## Post_test
##        n  missing distinct     Info     Mean      Gmd
##        9        0        6    0.958    59.56      3.5
##
## Value         56    59    60    61    62    63
## Frequency      3     1     1     1     1     2
## Proportion 0.333 0.111 0.111 0.111 0.111 0.222
##
## For the frequency table, variable is rounded to the nearest 0
## --------------------------------------------------------------------------------
```

```r
desc_stats_pastecs <- pastecs::stat.desc(df)
print(desc_stats_pastecs)
```

```
##                  Student     Pre_test    Post_test
## nbr.val        9.0000000   9.00000000   9.00000000
## nbr.null       0.0000000   0.00000000   0.00000000
## nbr.na         0.0000000   0.00000000   0.00000000
## min            1.0000000  47.00000000  56.00000000
## max           10.0000000  63.00000000  63.00000000
## range          9.0000000  16.00000000   7.00000000
## sum           48.0000000 499.00000000 536.00000000
## median         5.0000000  55.00000000  60.00000000
## mean           5.3333333  55.44444444  59.55555556
## SE.mean        1.1055416   1.61684802   0.98757716
## CI.mean.0.95   2.5493835   3.72845823   2.27735701
## var           11.0000000  23.52777778   8.77777778
## std.dev        3.3166248   4.85054407   2.96273147
## coef.var       0.6218671   0.08748476   0.04974736
```

2. The Department of Agriculture was studying the effects of several levels of a fertilizer on the growth of a plant. For some analyses, it might be useful to convert the fertilizer levels to an ordered factor.

```r
fertilizer_levels <- c("Low", "Medium", "High", "Low", "Medium", "High")

ordered_fertilizer <- factor(fertilizer_levels, ordered = TRUE, levels = c("Low", "Medium", "High"))

print(ordered_fertilizer)
```

```
## [1] Low    Medium High   Low    Medium High  
## Levels: Low < Medium < High
```

Figure 1: Student Score • The data were 10,10,10, 20,20,50,10,20,10,50,20,50,20,10. a. Write the codes and describe the result.

```r
exercise_levels <- c("l", "n", "n", "i", "l", "l", "n", "n", "i", "l")

exercise_factor <- factor(exercise_levels, levels = c("n", "l", "i"), labels = c("none", "light", "inte

print(exercise_factor)
```

```
##  [1] light   none    none    intense light   light   none    none    intense
## [10] light  
## Levels: none light intense
```

3. Abdul Hassan, president of Floor Coverings Unlimited, has asked you to study the exercise levels undertaken by 10 subjects were “l”, “n”, “n”, “i”, “l” , “l”, “n”, “n”, “i”, “l” ; n=none, l=light, i=intense

a. What is the best way to represent this in R?

```r
state <- c("tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", "qld",
           "vic", "nsw", "vic", "qld", "qld", "sa", "tas", "sa", "nt",
           "wa", "vic", "qld", "nsw", "nsw", "wa", "sa", "act", "nsw",
           "vic", "vic", "act")

state_factor <- factor(state)

print(levels(state_factor))
```

```
## [1] "act" "nsw" "nt"  "qld" "sa"  "tas" "vic" "wa"
```

4. Sample of 30 tax accountants from all the states and territories of Australia and their individual state of origin is specified by a character vector of state mnemonics as: state <- c("tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", "qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", "sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", "sa", "act", "nsw", "vic", "vic", "act")

a. Apply the factor function and factor level. Describe the results.

```r
custom_levels <- c("act", "nsw", "nt", "qld", "sa", "tas", "vic", "wa")

state_factor_custom <- factor(state, levels = custom_levels)

print(levels(state_factor_custom))
```

```
## [1] "act" "nsw" "nt"  "qld" "sa"  "tas" "vic" "wa"
```

5. From #4 - continuation: • Suppose we have the incomes of the same tax accountants in another vector (in suitably large units of money)

```r
incomes <- c(60, 49, 40, 61, 64, 60, 59, 54,
62, 69, 70, 42, 56, 61, 61, 61, 58, 51, 48,
65, 49, 49, 41, 48, 52, 46, 59, 46, 58, 43)

print(incomes)
```

```
##  [1] 60 49 40 61 64 60 59 54 62 69 70 42 56 61 61 61 58 51 48 65 49 49 41 48 52
## [26] 46 59 46 58 43
```

a. Calculate the sample mean income for each state we can now use the special function tapply(): Example: giving a means vector with the components labelled by the levels incmeans <- tapply(incomes, statef, mean)

Note: The function tapply() is used to apply a function, here mean(), to each group of components of the first argument, here incomes, defined by the levels of the second component, here state 2 • 2 that tapply() also works in this case when its second argument is not a factor, • e.g., 'tapply(incomes, state)', and this is true for quite a few other functions, since arguments are coerced to factors when necessary (using as.factor()).

```r
incmeans <- tapply(incomes, state_factor_custom, mean)

print(incmeans)
```

```
##      act      nsw       nt      qld       sa      tas      vic       wa
## 44.50000 57.33333 55.50000 53.60000 55.00000 60.50000 56.00000 52.25000
```

b. Copy the results and interpret.

Interpretation of the specific values in incmeans depends on the actual results, but in general, it represents the average income for tax accountants in each state based on the provided data.

6. Calculate the standard errors of the state income means (refer again to number 3) stdError <- function(x) sqrt(var(x)/length(x)) Note: After this assignment, the standard errors are calculated by: incster <- tapply(incomes, statef, stdError)

a. What is the standard error? Write the codes.

```
stdError <- function(x) sqrt(var(x) / length(x))

incster <- tapply(incomes, state_factor_custom, stdError)

print(incster)
```

```
##      act      nsw       nt      qld       sa      tas      vic       wa
## 1.500000 4.310195 4.500000 4.106093 2.738613 0.500000 5.244044 2.657536
```

b. Interpret the result.

#A smaller standard error indicates greater precision. In the context of this analysis, a smaller standard error for a state's mean income suggests that the sample mean is likely #more reliable estimate of the true mean income for tax accountants in that state.

7. Use the titanic dataset.

a. subset the titatic dataset of those who survived and not survived. Show the codes and its result.

```
library(titanic)
```

```
## Warning: package 'titanic' was built under R version 4.3.2
```

```
data("titanic_train")

survived_data <- subset(titanic_train, Survived == 1)
not_survived_data <- subset(titanic_train, Survived == 0)

print("Subset for those who survived:")
```

```
## [1] "Subset for those who survived:"
```

```
head(survived_data)
```

```
##    PassengerId Survived Pclass
## 2            2        1      1
## 3            3        1      3
## 4            4        1      1
## 9            9        1      3
## 10          10        1      2
## 11          11        1      3
##                                                   Name    Sex Age SibSp Parch
## 2  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                               Heikkinen, Miss. Laina female  26     0     0
## 4         Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 9    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27     0     2
## 10                 Nasser, Mrs. Nicholas (Adele Achem) female  14     1     0
## 11                     Sandstrom, Miss. Marguerite Rut female   4     1     1
##              Ticket    Fare Cabin Embarked
## 2          PC 17599 71.2833   C85        C
## 3  STON/O2. 3101282  7.9250              S
## 4            113803 53.1000  C123        S
## 9            347742 11.1333              S
## 10           237736 30.0708              C
## 11          PP 9549 16.7000    G6        S
```

```r
print("Subset for those who did not survive:")
```

```
## [1] "Subset for those who did not survive:"
```

```r
head(not_survived_data)
```

```
##    PassengerId Survived Pclass                           Name  Sex Age SibSp
## 1            1        0      3        Braund, Mr. Owen Harris male  22     1
## 5            5        0      3       Allen, Mr. William Henry male  35     0
## 6            6        0      3               Moran, Mr. James male  NA     0
## 7            7        0      1        McCarthy, Mr. Timothy J male  54     0
## 8            8        0      3 Palsson, Master. Gosta Leonard male   2     3
## 13          13        0      3 Saundercock, Mr. William Henry male  20     0
##    Parch    Ticket    Fare Cabin Embarked
## 1      0 A/5 21171  7.2500              S
## 5      0    373450  8.0500              S
## 6      0    330877  8.4583              Q
## 7      0     17463 51.8625   E46        S
## 8      1    349909 21.0750              S
## 13     0 A/5. 2151  8.0500              S
```