RWorksheet#6

Dianah Marie Canonicato

2023-12-06

install.packages("Hmisc") install.packages("pastecs")

#1

```r
install.packages("Hmisc")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
install.packages("pastecs")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'

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

```r
library(pastecs)
test <- data.frame(
  Student = c(1,2,3,4,5,6,7,8,9,10),
  PreTest = c(55,54,47,57,51,61,57,54,63,58),
  PostTest = c(61,60,56,63,56,63,59,56,62,61)
)
summary_hmisc <- describe(test)

summary_pastecs <- stat.desc(test)

cat("Descriptive Statistics using Hmisc:\n")
```

```
## Descriptive Statistics using Hmisc:
```

```r
print(summary_hmisc)
```

```
## test
##
##  3  Variables      10  Observations
## --------------------------------------------------------------------------------
## Student
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       10        0       10        1      5.5    3.667     1.45     1.90
##      .25      .50      .75      .90      .95
```

```
##     3.25     5.50     7.75     9.10     9.55
##
## Value         1    2    3    4    5    6    7    8    9   10
## Frequency     1    1    1    1    1    1    1    1    1    1
## Proportion 0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1
##
## For the frequency table, variable is rounded to the nearest 0
## ---------------------------------------------------------------------------
## PreTest
##        n  missing distinct     Info     Mean      Gmd
##       10        0        8    0.988     55.7    5.444
##
## Value       47   51   54   55   57   58   61   63
## Frequency    1    1    2    1    2    1    1    1
## Proportion 0.1  0.1  0.2  0.1  0.2  0.1  0.1  0.1
##
## For the frequency table, variable is rounded to the nearest 0
## ---------------------------------------------------------------------------
## PostTest
##        n  missing distinct     Info     Mean      Gmd
##       10        0        6    0.964     59.7    3.311
##
## Value       56   59   60   61   62   63
## Frequency    3    1    1    2    1    2
## Proportion 0.3  0.1  0.1  0.2  0.1  0.2
##
## For the frequency table, variable is rounded to the nearest 0
## ---------------------------------------------------------------------------
```

```
print(summary_pastecs)
```

```
##                  Student      PreTest     PostTest
## nbr.val       10.0000000  10.00000000  10.00000000
## 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           55.0000000 557.00000000 597.00000000
## median         5.5000000  56.00000000  60.50000000
## mean           5.5000000  55.70000000  59.70000000
## SE.mean        0.9574271   1.46855938   0.89504811
## CI.mean.0.95   2.1658506   3.32211213   2.02473948
## var            9.1666667  21.56666667   8.01111111
## std.dev        3.0276504   4.64399254   2.83039063
## coef.var       0.5504819   0.08337509   0.04741023
```

#2

```
Fertelizer_Data <- c(10, 10, 10, 20, 20, 50, 10, 20, 10, 50, 20, 50, 20, 10)
OrderedFertilizer <- factor(Fertelizer_Data, levels = c(10, 20, 50))

cat("Original data:\n")
```

```
## Original data:
```

```r
print(Fertelizer_Data)
```

```
##  [1] 10 10 10 20 20 50 10 20 10 50 20 50 20 10
```

```r
ordered_data <- OrderedFertilizer[order(OrderedFertilizer)]
cat("\nOrdered data:\n")
```

```
##
## Ordered data:
```

```r
print(ordered_data)
```

```
##  [1] 10 10 10 10 10 10 20 20 20 20 20 50 50 50
## Levels: 10 20 50
```

```r
#The result is the ordered version of the original data, with the values arranged in ascending order.
```

#3

```r
#Using a factor variable is the most effective method for expressing the workout levels in R.
#The three choices for activity levels in this instance are "n" (none), "l" (light), and "i" (intense).
#To appropriately describe the activity levels for each participant, you can use these levels to genera

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
```

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

#4

```r
States <- factor(state)
cat("Original state data:\n")
```

```
## Original state data:
```

```r
print(state)
```

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

```r
cat("\nFactor levels:\n")
```

```
##
## Factor levels:
```

```r
print(levels(States))
```

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

```r
cat("\nFactor representation:\n")
```

```
##
## Factor representation:
```

```r
print(States)
```

```
##  [1] tas sa  qld nsw nsw nt  wa  wa  qld vic nsw vic qld qld sa  tas sa  nt  wa 
## [20] vic qld nsw nsw wa  sa  act nsw vic vic act
## Levels: act nsw nt qld sa tas vic wa
```

```r
#The result will show the original state data, the factor levels, and the factor representation.
#The factor levels will be automatically assigned based on the unique values in the state vector.
```