In Class Assignment 3

Andrew Shao

2024-09-18

2.11.4 Q1

One advantage is it stores values as integers which takes less space. Another advantage is that it detects new input that is not in the levels which is useful for catching input mistakes.

2.11.4 Q2

C

2.11.4 Q3

A

2.11.4 Q4

B

2.11.4 Q5

```
size <- rep(c("big", "small", "medium"), 3:1)
size <- factor(size, ordered = T, levels = c('small', 'medium', 'big'))
size
```

```
## [1] big    big    big    small  small  medium
## Levels: small < medium < big
```

3.1.7 Q1

```
x <- diag(rep(1, 6)) + 1
x
```

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

3.1.7 Q2

```r
X <- matrix(1:16, 4, 4) + diag(4)
colMeans(X)
```

```
## [1]  2.75  6.75 10.75 14.75
```

3.1.7 Q3

```r
apply(X, 1, quantile, c(0.4, 0.7))
```

```
##     [,1] [,2] [,3] [,4]
## 40%  5.8  7.6  8.0  8.8
## 70%  9.4 10.4 12.3 12.5
```

3.1.7 Q4

```r
row_cumsum <- apply(X, 1, cumsum)
row_cumsum
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    2    3    4
## [2,]    7    9   10   12
## [3,]   16   19   22   24
## [4,]   29   33   37   41
```

```r
class(row_cumsum)
```

```
## [1] "matrix" "array"
```

The first column is the cumulative sums of the first row.

3.1.7 Q5

```r
b <- 1:4
a <- solve(X) %*% b
a
```

```
##            [,1]
## [1,]  1.2222222
## [2,]  0.6666667
## [3,]  0.1111111
## [4,] -0.4444444
```

3.3.4 Q1

```r
animal <- rep(c("sheep", "pig"), c(3,3))
weight <- c(110, NA, 140, NA, 300, 800)
condition <- c("excellent", "good", NA, "excellent", "good", "average")
healthy <- c(rep(TRUE, 5), FALSE)
my_data_frame <- data.frame(animal, weight, condition, healthy)
my_data_frame
```

```
##   animal weight condition healthy
## 1  sheep    110 excellent    TRUE
```

```
## 2  sheep     NA      good    TRUE
## 3  sheep    140      <NA>    TRUE
## 4    pig     NA excellent    TRUE
## 5    pig    300      good    TRUE
## 6    pig    800   average   FALSE
```

```
my_data_frame_nona <- na.omit(my_data_frame)
my_data_frame_nona
```

```
##   animal weight condition healthy
## 1  sheep    110 excellent    TRUE
## 5    pig    300      good    TRUE
## 6    pig    800   average   FALSE
```

3.3.4 Q2

```
my_data_frame[is.na(my_data_frame$weight), 'weight'] <- median(my_data_frame$weight,
↪  na.rm = T)
my_data_frame
```

```
##   animal weight condition healthy
## 1  sheep    110 excellent    TRUE
## 2  sheep    220      good    TRUE
## 3  sheep    140      <NA>    TRUE
## 4    pig    220 excellent    TRUE
## 5    pig    300      good    TRUE
## 6    pig    800   average   FALSE
```

3.3.4 Q3

```
my_data_frame <- rbind(my_data_frame, c('pig', 900, 'average', F))
my_data_frame
```

```
##   animal weight condition healthy
## 1  sheep    110 excellent    TRUE
## 2  sheep    220      good    TRUE
## 3  sheep    140      <NA>    TRUE
## 4    pig    220 excellent    TRUE
## 5    pig    300      good    TRUE
## 6    pig    800   average   FALSE
## 7    pig    900   average   FALSE
```

3.3.4 Q4

```
my_data_frame_sub <- my_data_frame[my_data_frame$weight < 400 & my_data_frame$condition
↪  %in% c('good', 'excellent'), c('animal', 'healthy')]
my_data_frame_sub
```

```
##   animal healthy
## 1  sheep    TRUE
## 2  sheep    TRUE
## 4    pig    TRUE
## 5    pig    TRUE
```