빅데이터 분석 시스템 개발_최현진

2021-07-02

0)(문제x)기본 세팅

```r
#0
setwd("C:/Users/bigdate/Desktop/workspace/R")
install.packages('dplyr')
library(dplyr)
```

```
> setwd("C:/Users/bigdate/Desktop/workspace/R")
> install.packages('dplyr')
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate
 version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/bigdate/Documents/R/win-library/3.6'
(as 'lib' is unspecified)

  There is a binary version available but the source version is
  later:
      binary source needs_compilation
dplyr  1.0.6  1.0.7              TRUE

  Binaries will be installed
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/dplyr_1.0.6.zip'
Content type 'application/zip' length 1559698 bytes (1.5 MB)
downloaded 1.5 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\bigdate\AppData\Local\Temp\RtmpOeEDB6\downloaded_packages
> library(dplyr)

다음의 패키지를 부착합니다: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

1) exam.csv 파일을 데이터 프레임 출력

```
#1
df_exam<-read.csv('./file/exam.csv')
View(df_exam)
```

```
> #1
> df_exam<-read.csv('./file/exam.csv')
> View(df_exam)
>
```

test07-02.R × df_exam ×

Filter

	id	class	math	english	science
1	1	1	50	98	50
2	2	1	60	97	60
3	3	1	45	86	78
4	4	1	30	98	58
5	5	2	25	80	65
6	6	2	50	89	98
7	7	2	80	90	45
8	8	2	90	78	25
9	9	3	20	98	15
10	10	3	50	98	45
11	11	3	65	65	65
12	12	3	45	85	32
13	13	4	46	98	65
14	14	4	48	87	12
15	15	4	75	56	78
16	16	4	58	98	65
17	17	5	65	68	98
18	18	5	80	78	90
19	19	5	89	68	87
20	20	5	78	83	58

2)math, english, science 변수만 갖는 데이터프레임 출력

```
#2
df_math<-df_exam %>% select(math)
df_english<-df_exam %>% select(english)
df_science<-df_exam %>% select(science)

df_rs1<-df_exam %>%  select(math)
df_rs2<-df_exam %>%  select(english)
df_rs3<-df_exam %>%  select(science)


df_math
df_english
df_science
```

```
> #2
> df_math<-df_exam %>% select(math)
> df_english<-df_exam %>% select(english)
> df_science<-df_exam %>% select(science)
> df_rs1<-df_exam %>%  select(math)
> df_rs2<-df_exam %>%  select(english)
> df_rs3<-df_exam %>%  select(science)
>
```

```
   math
1    50
2    60
3    45
4    30
5    25
6    50
7    80
8    90
9    20
10   50
11   65
12   45
13   46
14   48
15   75
16   58
17   65
18   80
19   89
20   78
>
```

```
> df_english
   english
1       98
2       97
3       86
4       98
5       80
6       89
7       90
8       78
9       98
10      98
11      65
12      85
13      98
14      87
15      56
16      98
17      68
18      78
19      68
20      83
```

```
> df_science
   science
1       50
2       60
3       78
4       58
5       65
6       98
7       45
8       25
9       15
10      45
11      65
12      32
13      65
14      12
15      78
16      65
17      98
18      90
19      87
20      58
```

3)Class가 1인 모든 변수를 갖는 데이터 프레임 출력

```
#3
df_rs4<-df_exam %>%  select(everything())
df_class1 <- df_exam %>% select(everything()) %>%  filter(class==1)
df_class1
```

```
> #3
> df_rs4<-df_exam %>%  select(everything())
> df_class1 <- df_exam %>% select(everything()) %>%  filter(class==1)
> df_class1
  id class math english science
1  1     1   50      98      50
2  2     1   60      97      60
3  3     1   45      86      78
4  4     1   30      98      58
> 
```

4)Math가 60점 이상 80점 미만 데이터 프레임 출력

```r
#4
df_math <- df_exam %>%  filter(math >= 60 & math < 80)
df_math
```

```
> #4
> df_math <- df_exam %>%  filter(math >= 60 & math < 80)
> df_math
  id class math english science
1  2     1   60      97      60
2 11     3   65      65      65
3 15     4   75      56      78
4 17     5   65      68      98
5 20     5   78      83      58
```

5)English가 60점 이상 80점 미만 데이터 프레임 출력

```
#5
df_english <- df_exam %>%  filter(english >= 60 & english <80)
df_english
```

```
> #5
> df_english <- df_exam %>%  filter(english >= 60 & english <80)
> df_english
  id class math english science
1  8     2   90      78      25
2 11     3   65      65      65
3 17     5   65      68      98
4 18     5   80      78      90
5 19     5   89      68      87
>
```

6)Math가 60점 이상이고 점수가 높은 순서를 갖는 class, id, math 변수를 갖는 데이터 프레임 출력

```r
#6
df_result <- df_exam %>% select(class, id, math) %>%
  filter(math >= 60) %>%
  arrange(asc(math)) %>%
  head(3)
df_result
```

```
Error: arrange() failed at implicit mutate() step.
* Problem with `mutate()` column `..1`.
i `..1 = asc(math)`.
x could not find function "asc"
Run `rlang::last_error()` to see where the error occurred.
> df_result
  class id math
1     2  8   90
2     5 19   89
3     2  7   80
```

7)Class로 그룹화되고 수학점수 평균(mean_math) 변수를 갖는 데이터 프레임 출력

```
#7
df_exam %>%  summarise(mean_math = mean(math))
```

```
> #7
> df_exam %>%  summarise(mean_math = mean(math))
  mean_math
1     57.45
```

8)total(math, english, science의 합) 파생변수를 갖는 데이터 프레임 출력

```
#8
df_total <- df_exam %>% mutate(total = math+english+science)
df_total
```

```
> #8
> df_total <- df_exam %>% mutate(total = math+english+science)
> df_total
   id class math english science total
1   1     1   50      98      50   198
2   2     1   60      97      60   217
3   3     1   45      86      78   209
4   4     1   30      98      58   186
5   5     2   25      80      65   170
6   6     2   50      89      98   237
7   7     2   80      90      45   215
8   8     2   90      78      25   193
9   9     3   20      98      15   133
10 10     3   50      98      45   193
11 11     3   65      65      65   195
12 12     3   45      85      32   162
13 13     4   46      98      65   209
14 14     4   48      87      12   147
15 15     4   75      56      78   209
16 16     4   58      98      65   221
17 17     5   65      68      98   231
18 18     5   80      78      90   248
19 19     5   89      68      87   244
20 20     5   78      83      58   219
```

9)mean(math, english, science의 합의 평균) 파생 변수를 갖는 데이터 프레임 출력

```
#9
df_mean <- df_total %>%  mutate(mean = total/3)
df_mean
```

```
> #9
> df_mean <- df_total %>%  mutate(mean = total/3)
> df_mean
   id class math english science total     mean
1   1     1   50      98      50   198 66.00000
2   2     1   60      97      60   217 72.33333
3   3     1   45      86      78   209 69.66667
4   4     1   30      98      58   186 62.00000
5   5     2   25      80      65   170 56.66667
6   6     2   50      89      98   237 79.00000
7   7     2   80      90      45   215 71.66667
8   8     2   90      78      25   193 64.33333
9   9     3   20      98      15   133 44.33333
10 10     3   50      98      45   193 64.33333
11 11     3   65      65      65   195 65.00000
12 12     3   45      85      32   162 54.00000
13 13     4   46      98      65   209 69.66667
14 14     4   48      87      12   147 49.00000
15 15     4   75      56      78   209 69.66667
16 16     4   58      98      65   221 73.66667
17 17     5   65      68      98   231 77.00000
18 18     5   80      78      90   248 82.66667
19 19     5   89      68      87   244 81.33333
20 20     5   78      83      58   219 73.00000
>
```

10)grade(평균의 등급, a,b,c,d,f)파생 변수를 갖는 데이터프레임 출력

```
#10
df_grade <- df_mean %>%  mutate(grade = ifelse(mean >= 90,'a',
                                               ifelse(mean >= 80,'b',
                                                      ifelse(mean >= 70,'c',
                                                             ifelse(mean >= 60,'d',
                                                                    'f')))))
df_grade
```

```
> #10
> df_grade <- df_mean %>%  mutate(grade = ifelse(mean >= 90,'a',
+                                                ifelse(mean >= 80,'b',
+                                                       ifelse(mean >= 70,'c',
+                                                              ifelse(mean >= 60,'d',
+                                                                     'f')))))
> df_grade
   id class math english science total     mean grade
1   1     1   50      98      50   198 66.00000     d
2   2     1   60      97      60   217 72.33333     c
3   3     1   45      86      78   209 69.66667     d
4   4     1   30      98      58   186 62.00000     d
5   5     2   25      80      65   170 56.66667     f
6   6     2   50      89      98   237 79.00000     c
7   7     2   80      90      45   215 71.66667     c
8   8     2   90      78      25   193 64.33333     d
9   9     3   20      98      15   133 44.33333     f
10 10     3   50      98      45   193 64.33333     d
11 11     3   65      65      65   195 65.00000     d
12 12     3   45      85      32   162 54.00000     f
13 13     4   46      98      65   209 69.66667     d
14 14     4   48      87      12   147 49.00000     f
15 15     4   75      56      78   209 69.66667     d
16 16     4   58      98      65   221 73.66667     c
17 17     5   65      68      98   231 77.00000     c
18 18     5   80      78      90   248 82.66667     b
19 19     5   89      68      87   244 81.33333     b
20 20     5   78      83      58   219 73.00000     c
```