

2

tidy 문법 익히기

▼ 목차

1. dplyr 패키지

filter()

1. 연속변수 처리

연습문제

2. 이산변수 처리

%in%

연습문제

3. 문자열 필터

pipeline '%>%' (매우중요)

select()

filter()와 select() 혼합

group_by() & summarise()

arrange를 이용한 데이터 정렬

정렬 이전 데이터

정렬 이후 데이터

결측치 처리

결측치 처리 방법 1

결측치 처리 방법 2

연습문제

mutate를 통한 변수(열) 생성

최종 연습문제

데이터 확인

연습문제

1. 비행 달이 7, 8, 9월인 행만 추려내시오.

2. 목적지(dest)가 "IAH" 이거나 "HOU"인 행만 추려내시오.

3. 도착지연 시간(arr_delay)이 60분 이고, 출발지연 시간(dep_delay)이 0분인 행만 추려내시오.

4. year, month, day 열만 추려내시오.

5. dep_time부터 arr_delay열까지 한꺼번에 추려내시오.

6. year, month, day 에 따른 dep_delay의 평균을 구하시오. (결측치도 처리할 것)

7. 목적지(dest)에 따른 dep_delay의 평균을 구해 내림차 순으로 정리하시오.

1. dplyr 패키지

filter()

`filter()` 는 각각 행(row)을 추려내는 역할

기본 내장 데이터인 mpg 데이터를 통해 실습합니다. (`따라한 뒤 제출`)

`glimpse()`

```
data(mpg)
glimpse(mpg)
```

```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 qua...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "f", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 1...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 2...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "comp..."
```

`head`

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans drv   cty   hwy fl   class
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi          a4     1.80  1999     4 auto... f     18    29 p   comp...
## 2 audi          a4     1.80  1999     4 manu... f     21    29 p   comp...
## 3 audi          a4     2.00  2008     4 manu... f     20    31 p   comp...
## 4 audi          a4     2.00  2008     4 auto... f     21    30 p   comp...
## 5 audi          a4     2.80  1999     6 auto... f     16    26 p   comp...
## 6 audi          a4     2.80  1999     6 manu... f     18    26 p   comp...
```



`glimpse()` 는 `head()` 보다 조금 더 자세하게 데이터를 보여줍니다.

1. 연속변수 처리

```
summary(mpg$cty)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   9.00  14.00   17.00   16.86  19.00   35.00
```

중앙값보다 큰 차들만 분류

```
filter(mpg, cty > 17)
```

```
## # A tibble: 102 x 11
##   manufacturer model   displ  year  cyl trans  drv      cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
## 1 audi          a4        1.80  1999    4 auto(l... f        18    29 p
## 2 audi          a4        1.80  1999    4 manual... f        21    29 p
## 3 audi          a4        2.00  2008    4 manual... f        20    31 p
## 4 audi          a4        2.00  2008    4 auto(a... f        21    30 p
## 5 audi          a4        2.80  1999    6 manual... f        18    26 p
## 6 audi          a4        3.10  2008    6 auto(a... f        18    27 p
## 7 audi          a4 quat... 1.80  1999    4 manual... 4        18    26 p
## 8 audi          a4 quat... 2.00  2008    4 manual... 4        20    28 p
## 9 audi          a4 quat... 2.00  2008    4 auto(s... 4        19    27 p
## 10 chevrolet    malibu    2.40  1999    4 auto(l... f        19    27 r
## # ... with 92 more rows, and 1 more variable: class <chr>
```

- filter 이후 기존 234개의 행에서 102개의 행으로 축소된 것을 확인할 수 있습니다.

연습문제

1. hwy가 10 이상인 것들만 추리시오
2. year가 2000년 이후인 것만 추리시오
3. cty가 10 미만이고, year가 2000년 미만인 것들만 추리시오. (&를 활용할 것)

2. 이산변수 처리

여기서 이산변수가 무엇이지를 잠시 살펴볼 필요가 있다.

앞서 나왔던 **hwy** 나 **cty** 등은 연속 변수

- 온도나 연비 등은 특정한 기준을 만든 것이 아니라, 그 값을 그저 '측정' 한 것들 (키, 몸무게도 마찬가지)

↔ **cyl** 이나 **displ** 같은 변수는 기준에 맞춰 숫자를 할당.

- 실린더 기통은 4개이거나 6개이거나 8개일 수밖에 없다.
- 이런 변수들은 나중에 factor로 분류해 처리할 수도 있다. (하지만 이 부분은 나중에)

%in%

cyl이 4인 경우만 추리려면, 어떻게 하면 될까?

→ `%in%` 를 통해서 변수의 내용을 추릴 수 있다. 아래를 살펴보자.

```
filter(mpg, cyl %in% 4)
```

```
## # A tibble: 81 x 11
##   manufacturer model    displ  year  cyl trans  drv    cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
## 1 audi          a4        1.80  1999    4 auto(l... f      18    29 p
## 2 audi          a4        1.80  1999    4 manual... f      21    29 p
## 3 audi          a4        2.00  2008    4 manual... f      20    31 p
## 4 audi          a4        2.00  2008    4 auto(a... f      21    30 p
## 5 audi          a4 quat... 1.80  1999    4 manual... 4      18    26 p
## 6 audi          a4 quat... 1.80  1999    4 auto(l... 4      16    25 p
## 7 audi          a4 quat... 2.00  2008    4 manual... 4      20    28 p
## 8 audi          a4 quat... 2.00  2008    4 auto(s... 4      19    27 p
## 9 chevrolet     malibu    2.40  1999    4 auto(l... f      19    27 r
## 10 chevrolet    malibu    2.40  2008    4 auto(l... f      22    30 r
## # ... with 71 more rows, and 1 more variable: class <chr>
```

`%in%` 를 여러 개의 값에 대해서도 사용 가능

```
filter(mpg, cyl %in% c(4, 6))
```

```
## # A tibble: 160 x 11
##   manufacturer model    displ  year   cyl trans  drv    cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
## 1 audi          a4        1.80  1999     4 auto(l... f      18    29 p
## 2 audi          a4        1.80  1999     4 manual... f      21    29 p
## 3 audi          a4        2.00  2008     4 manual... f      20    31 p
## 4 audi          a4        2.00  2008     4 auto(a... f      21    30 p
## 5 audi          a4        2.80  1999     6 auto(l... f      16    26 p
## 6 audi          a4        2.80  1999     6 manual... f      18    26 p
## 7 audi          a4        3.10  2008     6 auto(a... f      18    27 p
## 8 audi          a4 quat... 1.80  1999     4 manual... 4      18    26 p
## 9 audi          a4 quat... 1.80  1999     4 auto(l... 4      16    25 p
## 10 audi         a4 quat... 2.00  2008     4 manual... 4      20    28 p
## # ... with 150 more rows, and 1 more variable: class <chr>
```

연습문제

1. displ이 1.8인 경우만 추려내시오.
2. displ이 2.0이고 cyl이 6, 8인 경우만 추려내시오.

3. 문자열 필터

| 문자열도 위와 동일한 원리

```
filter(mpg, class == "suv")
```

```
## # A tibble: 62 x 11
##   manufacturer model      displ  year   cyl trans drv      cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
## 1 chevrolet    c1500 sub...  5.30  2008     8 auto... r      14    20 r
## 2 chevrolet    c1500 sub...  5.30  2008     8 auto... r      11    15 e
## 3 chevrolet    c1500 sub...  5.30  2008     8 auto... r      14    20 r
## 4 chevrolet    c1500 sub...  5.70  1999     8 auto... r      13    17 r
## 5 chevrolet    c1500 sub...  6.00  2008     8 auto... r      12    17 r
## 6 chevrolet    k1500 tah...  5.30  2008     8 auto... 4      14    19 r
## 7 chevrolet    k1500 tah...  5.30  2008     8 auto... 4      11    14 e
## 8 chevrolet    k1500 tah...  5.70  1999     8 auto... 4      11    15 r
## 9 chevrolet    k1500 tah...  6.50  1999     8 auto... 4      14    17 d
## 10 dodge       durango 4...  3.90  1999     6 auto... 4      13    17 r
## # ... with 52 more rows, and 1 more variable: class <chr>
```

여러 조건을 줄 때는 **&** 나 **|** 를 사용하면 된다.

```
filter(mpg, manufacturer == "hyundai" & hwy < 25)
```

```
## # A tibble: 3 x 11
##   manufacturer model displ  year   cyl trans drv      cty   hwy fl   class
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 hyundai      tibu...  2.70  2008     6 auto... f      17    24 r   subc...
## 2 hyundai      tibu...  2.70  2008     6 manu... f      16    24 r   subc...
## 3 hyundai      tibu...  2.70  2008     6 manu... f      17    24 r   subc...
```

문자열에서 **%in%** 를 사용하여 filtering한 예시


```
filter(mpg, year > 2000 & class %in% c("subcompact", "suv")) #2000년 이후, subcompact이거나 suv인 차량
```

```
## # A tibble: 49 x 11
##   manufacturer model      displ  year   cyl trans  drv      cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
## 1 chevrolet    c1500 sub...  5.30  2008     8 auto... r       14    20 r
## 2 chevrolet    c1500 sub...  5.30  2008     8 auto... r       11    15 e
## 3 chevrolet    c1500 sub...  5.30  2008     8 auto... r       14    20 r
## 4 chevrolet    c1500 sub...  6.00  2008     8 auto... r       12    17 r
## 5 chevrolet    k1500 tah...  5.30  2008     8 auto... 4       14    19 r
## 6 chevrolet    k1500 tah...  5.30  2008     8 auto... 4       11    14 e
## 7 dodge        durango 4...  4.70  2008     8 auto... 4       13    17 r
## 8 dodge        durango 4...  4.70  2008     8 auto... 4        9    12 e
## 9 dodge        durango 4...  4.70  2008     8 auto... 4       13    17 r
## 10 dodge       durango 4...  5.70  2008     8 auto... 4       13    18 r
## # ... with 39 more rows, and 1 more variable: class <chr>
```

```
filter(mpg, drv %in% "r" | class %in% "suv")
```

```
## # A tibble: 76 x 11
##   manufacturer model      displ  year   cyl trans  drv      cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
## 1 chevrolet    c1500 su...  5.30  2008     8 auto(... r       14    20 r
## 2 chevrolet    c1500 su...  5.30  2008     8 auto(... r       11    15 e
## 3 chevrolet    c1500 su...  5.30  2008     8 auto(... r       14    20 r
## 4 chevrolet    c1500 su...  5.70  1999     8 auto(... r       13    17 r
## 5 chevrolet    c1500 su...  6.00  2008     8 auto(... r       12    17 r
## 6 chevrolet    corvette   5.70  1999     8 manua... r       16    26 p
## 7 chevrolet    corvette   5.70  1999     8 auto(... r       15    23 p
## 8 chevrolet    corvette   6.20  2008     8 manua... r       16    26 p
## 9 chevrolet    corvette   6.20  2008     8 auto(... r       15    25 p
## 10 chevrolet    corvette   7.00  2008     8 manua... r       15    24 p
## # ... with 66 more rows, and 1 more variable: class <chr>
```


pipeline '%>%' (매우중요)

사실상 `tidyverse` 를 배우는 이유라고 할 수 있는 꽃  과도 같은 존재

귀찮게 치지 않고, 다음과 같은 단축키로 치기를 권장

- Window : Ctrl + Shift + M
- Mac : Cmd + Shift + M

```
mpg %>%  
  filter(year > 2000 & class %in% c("subcompact"))
```

```
## # A tibble: 16 x 11  
##   manufacturer model   displ  year  cyl trans  drv      cty   hwy fl  
##   <chr>          <chr>   <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>  
## 1 ford          mustang  4.00  2008    6 manual r      17    26 r  
## 2 ford          mustang  4.00  2008    6 auto(l r      16    24 r  
## 3 ford          mustang  4.60  2008    8 manual r      15    23 r  
## 4 ford          mustang  4.60  2008    8 auto(l r      15    22 r  
## 5 ford          mustang  5.40  2008    8 manual r      14    20 p  
## 6 honda         civic    1.80  2008    4 manual f      26    34 r  
## 7 honda         civic    1.80  2008    4 auto(l f      25    36 r  
## 8 honda         civic    1.80  2008    4 auto(l f      24    36 c  
## 9 honda         civic    2.00  2008    4 manual f      21    29 p  
## 10 hyundai       tiburon  2.00  2008    4 manual f      20    28 r  
## 11 hyundai       tiburon  2.00  2008    4 auto(l f      20    27 r  
## 12 hyundai       tiburon  2.70  2008    6 auto(l f      17    24 r  
## 13 hyundai       tiburon  2.70  2008    6 manual f      16    24 r  
## 14 hyundai       tiburon  2.70  2008    6 manual f      17    24 r  
## 15 volkswagen    new bee  2.50  2008    5 manual f      20    28 r  
## 16 volkswagen    new bee  2.50  2008    5 auto(s f      20    29 r  
## # ... with 1 more variable: class <chr>
```



코드의 가독성이 기가 막히지 않나요? (아님 말구요)

select()

```
mpg %>%  
  select(model, year, class)
```

```
## # A tibble: 234 x 3  
##   model      year class  
##   <chr>    <int> <chr>  
## 1 a4        1999 compact  
## 2 a4        1999 compact  
## 3 a4        2008 compact  
## 4 a4        2008 compact  
## 5 a4        1999 compact  
## 6 a4        1999 compact  
## 7 a4        2008 compact  
## 8 a4 quattro 1999 compact  
## 9 a4 quattro 1999 compact  
## 10 a4 quattro 2008 compact  
## # ... with 224 more rows
```

- 3개의 열만 보고 싶을 때는, 열 이름 3개를 입력하는 게 그리 귀찮진 않다.

But, 보고 싶은 열의 수가 늘어난다면?

→ 3가지 방법이 있다.

3가지 방법

1. 보기 싫은 열을 "-"로 제외 - 예: mpg %>% select(-hwy, -fl, -class)

```
mpg %>% select(-hwy, -fl, -class)
```

```
## # A tibble: 234 x 8
##   manufacturer model      displ  year  cyl trans      drv    cty
##   <chr>          <chr>    <dbl> <int> <int> <chr>    <chr> <int>
## 1 audi          a4        1.80  1999    4 auto(l5)  f      18
## 2 audi          a4        1.80  1999    4 manual(m5) f      21
## 3 audi          a4        2.00  2008    4 manual(m6) f      20
## 4 audi          a4        2.00  2008    4 auto(av)   f      21
## 5 audi          a4        2.80  1999    6 auto(l5)  f      16
## 6 audi          a4        2.80  1999    6 manual(m5) f      18
## 7 audi          a4        3.10  2008    6 auto(av)   f      18
## 8 audi          a4 quattro  1.80  1999    4 manual(m5) 4      18
## 9 audi          a4 quattro  1.80  1999    4 auto(l5)   4      16
## 10 audi         a4 quattro  2.00  2008    4 manual(m6) 4      20
## # ... with 224 more rows
```

2. 보고 싶은 열을 ":"로 묶기 - 예 : mpg %>% select(manufacturer:cty)

```
mpg %>% select(manufacturer:cty)
```

```
## # A tibble: 234 x 8
##   manufacturer model      displ  year  cyl trans      drv    cty
##   <chr>          <chr>    <dbl> <int> <int> <chr>    <chr> <int>
## 1 audi          a4        1.80  1999    4 auto(l5)  f      18
## 2 audi          a4        1.80  1999    4 manual(m5) f      21
## 3 audi          a4        2.00  2008    4 manual(m6) f      20
## 4 audi          a4        2.00  2008    4 auto(av)   f      21
## 5 audi          a4        2.80  1999    6 auto(l5)  f      16
## 6 audi          a4        2.80  1999    6 manual(m5) f      18
## 7 audi          a4        3.10  2008    6 auto(av)   f      18
## 8 audi          a4 quattro  1.80  1999    4 manual(m5) 4      18
## 9 audi          a4 quattro  1.80  1999    4 auto(l5)   4      16
## 10 audi         a4 quattro  2.00  2008    4 manual(m6) 4      20
## # ... with 224 more rows
```

3. 보기 싫은 열을 "-:"로 묶어서 제외 - 예: mpg %>% select(-hwy:-class)

```
mpg %>% select(-hwy:-class)
```

```
## # A tibble: 234 x 8
##   manufacturer model      displ  year   cyl trans      drv    cty
##   <chr>          <chr>    <dbl> <int> <int> <chr>    <chr> <int>
## 1 audi          a4        1.80  1999     4 auto(l5)  f      18
## 2 audi          a4        1.80  1999     4 manual(m5) f      21
## 3 audi          a4        2.00  2008     4 manual(m6) f      20
## 4 audi          a4        2.00  2008     4 auto(av)   f      21
## 5 audi          a4        2.80  1999     6 auto(l5)  f      16
## 6 audi          a4        2.80  1999     6 manual(m5) f      18
## 7 audi          a4        3.10  2008     6 auto(av)   f      18
## 8 audi          a4 quattro  1.80  1999     4 manual(m5) 4      18
## 9 audi          a4 quattro  1.80  1999     4 auto(l5)   4      16
## 10 audi         a4 quattro  2.00  2008     4 manual(m6) 4      20
## # ... with 224 more rows
```

filter()와 select() 혼합

파이프라인은 2개 이상 계속 쓸 수 있습니다.

- 갈때기에 한 번 걸러진 채로, 두 번 걸러지는 느낌이라고 생각하시면 쉽습니다.

```
mpg %>%
  filter(cty > 20) %>%
  select(model, year, cty:class)
```

```
## # A tibble: 45 x 6
##   model  year  cty  hwy fl  class
##   <chr> <int> <int> <int> <chr> <chr>
## 1 a4      1999   21   29 p  compact
## 2 a4      2008   21   30 p  compact
## 3 malibu  2008   22   30 r  midsize
## 4 civic   1999   28   33 r  subcompact
## 5 civic   1999   24   32 r  subcompact
## 6 civic   1999   25   32 r  subcompact
## 7 civic   1999   23   29 p  subcompact
## 8 civic   1999   24   32 r  subcompact
## 9 civic   2008   26   34 r  subcompact
## 10 civic  2008   25   36 r  subcompact
## # ... with 35 more rows
```

group_by() & summarise()

언급한 차종에 따른 고속도로 연비 평균을 구해보자.

우선 실행해야 하는 것은 `group_by()` 이다.

`group_by(기준 변수)` 를 실행한 후 결과를 보자.

```
mpg %>%  
  group_by(model)
```

```
## # A tibble: 234 x 11  
## # Groups:   model [38]  
##   manufacturer model   displ  year  cyl trans  drv    cty   hwy fl  
##   <chr>          <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr>  
## 1 audi          a4        1.80  1999    4 auto(l... f      18    29 p  
## 2 audi          a4        1.80  1999    4 manual... f      21    29 p  
## 3 audi          a4        2.00  2008    4 manual... f      20    31 p  
## 4 audi          a4        2.00  2008    4 auto(a... f      21    30 p  
## 5 audi          a4        2.80  1999    6 auto(l... f      16    26 p  
## 6 audi          a4        2.80  1999    6 manual... f      18    26 p  
## 7 audi          a4        3.10  2008    6 auto(a... f      18    27 p  
## 8 audi          a4 quat... 1.80  1999    4 manual... 4      18    26 p  
## 9 audi          a4 quat... 1.80  1999    4 auto(l... 4      16    25 p  
## 10 audi         a4 quat... 2.00  2008    4 manual... 4      20    28 p  
## # ... with 224 more rows, and 1 more variable: class <chr>
```



결과를 보면 "Groups: model [38]"이라는 한 줄이 추가되었을 뿐, **특별한 변화가 없다.**

하지만 이건 기준을 잡았으니, 다음 연산을 하라는 이야기다. 이제 `summarise()` 를 통해 고속도로 연비(hwy)의 평균을 구해보자.

즉, `group_by()` 와 `summarise()` 는 짝꿍입니다.

- `group_by()` 혼자서는 큰 의미 X

```
mpg %>%
  group_by(model) %>%
  summarise(hwy_mean = mean(hwy))
```

```
## # A tibble: 38 x 2
##   model          hwy_mean
##   <chr>          <dbl>
## 1 4runner 4wd      18.8
## 2 a4              28.3
## 3 a4 quattro      25.8
## 4 a6 quattro      24.0
## 5 altima          28.7
## 6 c1500 suburban 2wd 17.8
## 7 camry           28.3
## 8 camry solara     28.1
## 9 caravan 2wd      22.4
## 10 civic           32.6
## # ... with 28 more rows
```



이러한 코드 폼을 우리가 가장 많이 쓰게 될 겁니다.

- `%>%` 하나 당 엔터를 한 번씩 눌러서 가독성을 높여주는 게 좋습니다.
(최대한 위 코드폼과 동일하게 쳐 버릇해주세요)

arrange를 이용한 데이터 정렬

정렬 이전 데이터

```
mpg %>%
  group_by(model) %>%
  summarise(hwy_mean = mean(hwy), hwy_sum = sum(hwy))
```

```
## # A tibble: 38 x 3
##   model          hwy_mean hwy_sum
##   <chr>          <dbl>   <int>
## 1 4runner 4wd      18.8     113
## 2 a4             28.3     198
## 3 a4 quattro      25.8     206
## 4 a6 quattro      24.0      72
## 5 altima          28.7     172
## 6 c1500 suburban 2wd 17.8      89
## 7 camry           28.3     198
## 8 camry solara     28.1     197
## 9 caravan 2wd      22.4     246
## 10 civic           32.6     293
## # ... with 28 more rows
```

정렬 이후 데이터


```
mpg %>%
  group_by(model) %>%
  summarise(hwy_mean = mean(hwy), hwy_sum = sum(hwy)) %>%
  arrange(hwy_mean)
```

```
## # A tibble: 38 x 3
##   model                hwy_mean hwy_sum
##   <chr>                <dbl>   <int>
## 1 ram 1500 pickup 4wd    15.3     153
## 2 durango 4wd           16.0     112
## 3 k1500 tahoe 4wd       16.2      65
## 4 f150 pickup 4wd       16.4     115
## 5 land cruiser wagon 4wd 16.5      33
## 6 range rover           16.5      66
## 7 dakota pickup 4wd     17.0     153
## 8 navigator 2wd         17.0      51
## 9 expedition 2wd        17.3      52
## 10 grand cherokee 4wd    17.6     141
## # ... with 28 more rows
```

```
mpg %>%
  group_by(model) %>%
  summarise(hwy_mean = mean(hwy), hwy_sum = sum(hwy)) %>%
  arrange(desc(hwy_mean))
```

```
## # A tibble: 38 x 3
##   model      hwy_mean hwy_sum
##   <chr>      <dbl>   <int>
## 1 corolla      34.0     170
## 2 new beetle   32.8     197
## 3 civic        32.6     293
## 4 jetta        29.1     262
## 5 altima       28.7     172
## 6 a4           28.3     198
## 7 camry        28.3     198
## 8 camry solara 28.1     197
## 9 sonata       27.7     194
## 10 malibu      27.6     138
## # ... with 28 more rows
```

desc 옵션을 추가하면, 내림차순으로 sorting할 수 있습니다.

결측치 처리

혹시 `summarise()` 를 실행하다가 결과에 NA가 뜨고 계산이 안 될 때가 있다.

그럴 때는 데이터를 불러온 행의 처음에 `filter(!is.na(변수))` 를 써주면 문제가 해결

- 혹은 `summarise()` 에 쓰는 `sum()` 함수나 `mean()` 함수에 "`na.rm = TRUE`"를 추가한다.

결측치 처리 방법 1

```
nycflights13::flights %>%
  filter(!is.na(dep_delay)) %>%
  group_by(month) %>%
  summarise(delay_mean = mean(dep_delay))
```

```
## # A tibble: 12 x 2
##   month delay_mean
##   <int>     <dbl>
## 1     1      10.0
## 2     2      10.8
## 3     3      13.2
## 4     4      13.9
## 5     5      13.0
## 6     6      20.8
## 7     7      21.7
## 8     8      12.6
## 9     9       6.72
## 10    10       6.24
## 11    11       5.44
## 12    12      16.6
```

결측치 처리 방법 2

```
nycflights13::flights %>%
  group_by(month) %>%
  summarise(delay_mean = mean(dep_delay, na.rm = TRUE))
```

```
## # A tibble: 12 x 2
##   month delay_mean
##   <int>     <dbl>
## 1     1      10.0
## 2     2      10.8
## 3     3      13.2
## 4     4      13.9
## 5     5      13.0
## 6     6      20.8
## 7     7      21.7
## 8     8      12.6
## 9     9       6.72
## 10    10       6.24
## 11    11       5.44
## 12    12      16.6
```

연습문제

1. mpg 데이터의 year에 따른 hwy, cty의 평균을 구하고, cty_mean 기준으로 내림차순 정렬하시오.
2. mpg 데이터의 class에 따른 cty, hwy의 합을 구하고, hwy_sum 기준으로 오름차순 정렬하시오.

mutate를 통한 변수(열) 생성

hwy를 cty로 나눈, 즉 hwy/cty 를 시내 대비 고속도로 연비 (hwy_per_cty)라고 해보자.

mpg 데이터에 이러한 항목을 추가하려면 어떻게 해야 할까?

```
mpg %>%
  mutate(hwy_per_cty = hwy / cty)
```

```
## # A tibble: 234 x 12
##   manufacturer model   displ  year  cyl trans  drv    cty   hwy fl
##   <chr>          <chr>    <dbl> <int> <int> <chr>  <chr> <int> <int> <chr>
## 1 audi          a4        1.80  1999    4 auto(l... f      18    29 p
## 2 audi          a4        1.80  1999    4 manual... f      21    29 p
## 3 audi          a4        2.00  2008    4 manual... f      20    31 p
## 4 audi          a4        2.00  2008    4 auto(a... f      21    30 p
## 5 audi          a4        2.80  1999    6 auto(l... f      16    26 p
## 6 audi          a4        2.80  1999    6 manual... f      18    26 p
## 7 audi          a4        3.10  2008    6 auto(a... f      18    27 p
## 8 audi          a4 quat... 1.80  1999    4 manual... 4      18    26 p
## 9 audi          a4 quat... 1.80  1999    4 auto(l... 4      16    25 p
## 10 audi          a4 quat... 2.00  2008    4 manual... 4      20    28 p
## # ... with 224 more rows, and 2 more variables: class <chr>,
## #   hwy_per_cty <dbl>
```



아주 간단하죵

최종 연습문제

| 연습문제 풀이 전에 시행

1. 비행기 데이터 문제(nycflights13)
2. "nycflights13" 패키지를 설치한다. (install.packages("nycflights13"))
3. library(nycflights13) 명령어로 패키지를 로드하고, flights 데이터를 불러온다. (data(flights))

데이터 확인

```
library(nycflights13)
data(flights)

glimpse(flights)
```

연습문제

1. 비행 달이 7, 8, 9월인 행만 추려내시오.
2. 목적지(dest)가 "IAH" 이거나 "HOU"인 행만 추려내시오.
3. 도착지연 시간(arr_delay)이 60분 이고, 출발지연 시간(dep_delay)이 0분인 행만 추려내시오.
4. year, month, day 열만 추려내시오.
5. dep_time부터 arr_delay열까지 한꺼번에 추려내시오.
6. year, month, day 에 따른 dep_delay의 평균을 구하시오. (결측치도 처리할 것)
7. 목적지(dest)에 따른 dep_delay의 평균을 구해 내림차 순으로 정리하시오.

[맨위로](#)