공유자전거 수요 예측 분석

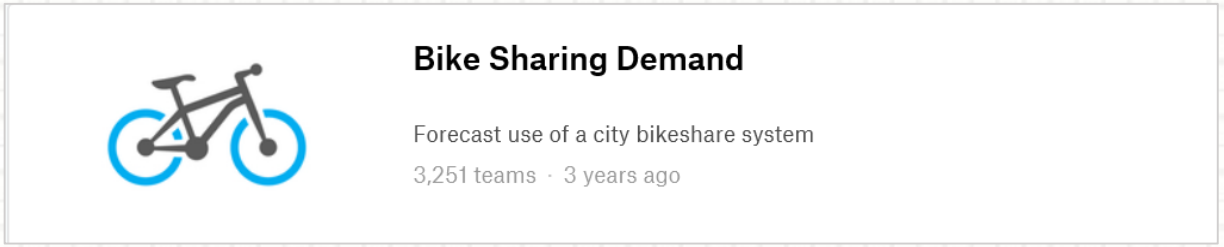
김석준 김성표 박민혁 이은진

INDEX

데이터 획득

01 데이터 획득

국외 자전거 수요 데이터 : Kaggle의 Bike Sharing Demand 데이터 (2011~2012년)



datetime	season	holiday	workingda	weather	temp	atemp	humidity	windspeed	casual	registered	count
2011-01-01 0:00	1	0	0	1	9.84	14.395	81	0	3	13	16
2011-01-01 1:00	1	0	0	1	9.02	13.635	80	0	8	32	40
2011-01-01 2:00	1	0	0	1	9.02	13.635	80	0	5	27	32
2011-01-01 3:00	1	0	0	1	9.84	14.395	75	0	3	10	13
2011-01-01 4:00	1	0	0	1	9.84	14.395	75	0	0	1	1
2011-01-01 5:00	1	0	0	2	9.84	12.88	75	6.0032	0	1	1
2011-01-01 6:00	1	0	0	1	9.02	13.635	80	0	2	0	2
2011-01-01 7:00	1	0	0	1	8.2	12.88	86	0	1	2	3
2011-01-01 8:00	1	0	0	1	9.84	14.395	75	0	1	7	8
2011-01-01 9:00	1	0	0	1	13.12	17.425	76	0	8	6	14
2011-01-01 10:00	1	0	0	1	15.58	19.695	76	16.9979	12	24	36
2011-01-01 11:00	1	0	0	1	14.76	16.665	81	19.0012	26	30	56
2011-01-01 12:00	1	0	0	1	17.22	21.21	77	19.0012	29	55	84
2011-01-01 13:00	1	0	0	2	18.86	22.725	72	19.9995	47	47	94
2011-01-01 14:00	1	0	0	2	18.86	22.725	72	19.0012	35	71	106
2011-01-01 15:00	1	0	0	2	18.04	21.97	77	19.9995	40	70	110
2011-01-01 16:00	1	0	0	2	17.22	21.21	82	19.9995	41	52	93

워싱턴 D.C 기준

- Datatime: 날짜
- season: 계절 (봄–1, 여름–2, 가을–3, 겨울–4)
- Hoilday: 공휴일(일반–0, 공휴일–1)
- Workingday:평일
- Weather: 날씨(좋음–1, 안개–2, 비–3, 많은 비–4)
- Temp: 기온
- Atemp: 체감온도
- Humidity: 습도
- Wkdspeed: 풍속(mile)
- Casual: 정기권 등록하지 않은 이용자 수
- Registered: 정기권을 등록한 이용자 수
- Count: 일별 총이용자 수

01 데이터 획득

국내 자전거 수요 데이터 : 서울시 정보소통광장 따릉이 자전거 이용 추이 데이터 & 서울시 기상청 날씨 데이터 (2015~2016년)

datetime	count	temp	humidity	weather	windspeed	atemp	season
2015-09-19	724	22.3	58.3	1	5.00	23.7088995	3
2015-09-20	907	22.1	69.0	2	3.75	23.6337524	3
2015-09-21	553	22.6	62.1	2	4.50	24.1013350	3
2015-09-22	796	23.9	48.4	1	4.25	25.5869076	3
2015-09-23	804	22.0	57.9	1	4.75	23.3965524	3
2015-09-24	914	23.4	56.4	1	6.25	24.8584303	3
2015-09-25	1022	23.4	56.5	2	4.50	25.0020384	3
2015-09-26	838	22.4	54.5	2	4.75	23.8486570	3
2015-09-27	706	21.9	39.4	1	4.00	23.3770227	3
2015-09-28	1382	21.4	39.6	3	5.75	22.6034914	3
2015-09-29	1870	22.1	36.6	1	6.50	23.3373960	3
2015-09-30	1087	21.3	45.3	1	9.00	22.2013537	3
2015-10-01	240	16.3	77.5	4	9.25	16.2455303	3
2015-10-02	1138	16.5	53.5	1	9.50	16.4535635	3
2015-10-03	32	19.2	49.4	1	4.50	20.2733457	3
2015-10-04	126	16.6	38.9	1	4.75	17.2931407	3
2015-10-05	204	18.0	49.4	1	4.75	18.8755067	3
2015-10-06	917	18.7	57.5	1	3.50	19.9087162	3
2015-10-07	923	19.6	67.5	1	5.75	20.5405946	3
2015-10-08	1106	20.1	61.1	1	9.25	20.7580771	3
2015-10-09	1504	16.2	43.8	1	5.50	16.6903913	3

서울특별시 기준

- datetime : 날짜
- season : (봄–1, 여름–2, 가을–3, 겨울–4)
- weather: (좋음–1, 안개–2, 비–3, 많은 비–4)
- temp: 기온
- atemp : 체감온도
- Humidity : 습도
- windspeed : 풍속(mile) (미국식으로 변환)
- count : 일별 총이용자 수

시 각 화

02-1 정식등록과 일일대여 자전거의 working day 여부에 따른 이용시간 추이

전처리: datetime에서 시간 추출 및 데이터형 변환

train$Hour <- as.factor(hour(ymd_hms(train$datetime)))

```
> str(train)
'data.frame':   10886 obs. of  12 variables:
 $ datetime  : chr  "2011-01-01 00:00:00" "2011-01-01 01:00:00"
"2011-01-01 03:00:00" ...
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weather   : int  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  9.84 9.02 9.02 9.84 9.84 ...
 $ atemp     : num  14.4 13.6 13.6 14.4 14.4 ...
 $ humidity  : int  81 80 80 75 75 75 80 86 75 76 ...
 $ windspeed : num  0 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ count     : int  16 40 32 13 1 1 2 3 8 14 ...
```

```
0" "2011-01-01 02:00:00" "2011-01-01 03:00:00" ...
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weather   : int  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  9.84 9.02 9.02 9.84 9.84 ...
 $ atemp     : num  14.4 13.6 13.6 14.4 14.4 ...
 $ humidity  : int  81 80 80 75 75 75 80 86 75 76 ...
 $ windspeed : num  0 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ count     : int  16 40 32 13 1 1 2 3 8 14 ...
 $ Hour      : Factor w/ 24 levels "0","1","2","3",..: 1 2 3
4 5 6 7 8 9 10 ...
>
```

02-1 정식등록과 일일대여 자전거의 working day 여부에 따른 이용시간 추이

전처리: 시간대별 자전거 이용시간 합산

1) Rent_type_RentHour <- aggregate(train[,c("정기권X", "정기권O")], by=list(train$Hour), "sum")

2) Rent_type_RentHour <- melt(Rent_type_RentHour[,c(('시간','정기권X','정기권O')], id.vars = 1)

1)

```
> Rent_type_RentHour
   Group.1 정기권X 정기권O
1        0    4692   20396
2        1    2957   12415
3        2    2159    8100
4        3    1161    3930
5        4     558    2274
6        5     658    8277
7        6    1888   32810
8        7    4966   92002
9        8    9802  155258
10       9   14085   86825
11      10   20984   58683
12      11   27324   68533
13      12   31387   85581
14      13   33771   83780
15      14   34925   76085
16      15   34669   81291
17      16   34238  110028
18      17   34401  179356
```

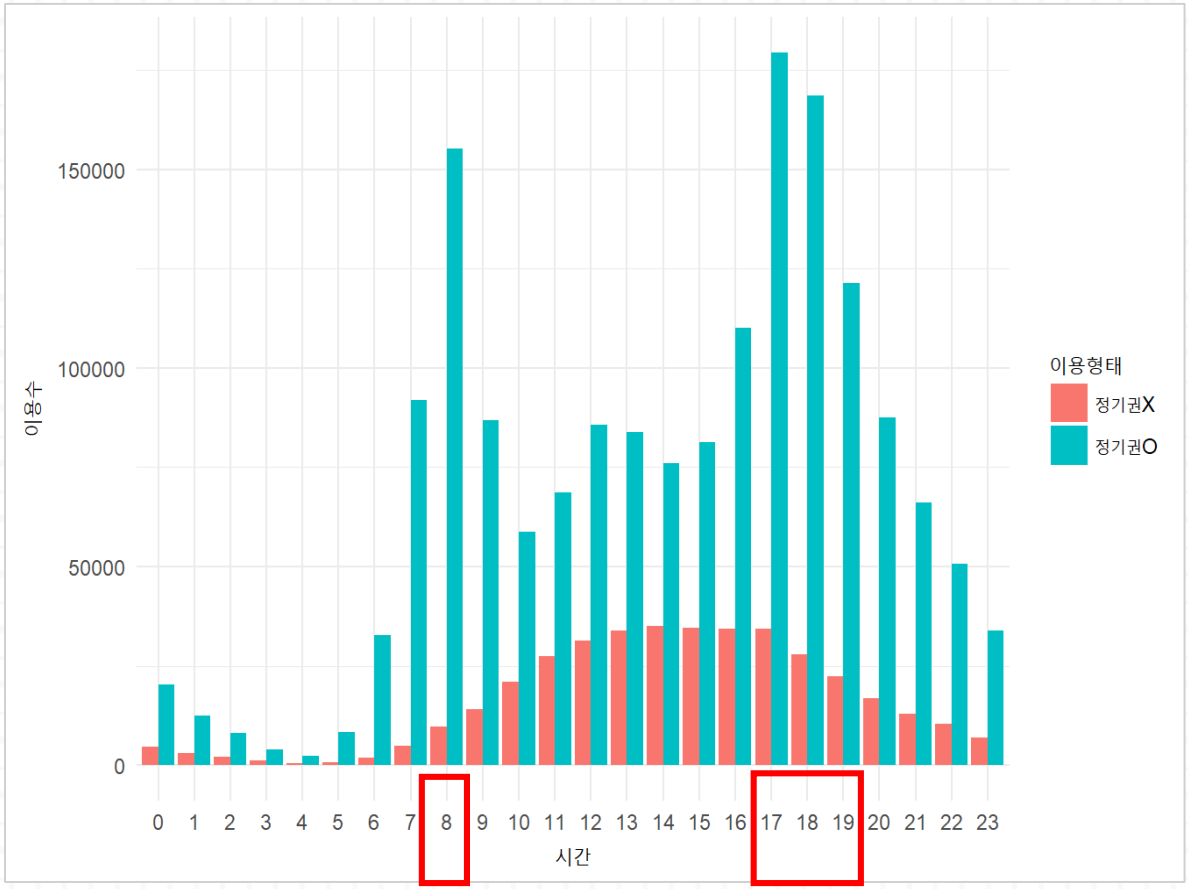
2)

```
> Rent_type_RentHour
   시간 이용형태 이용수
1     0  정기권X   4692
2     1  정기권X   2957
3     2  정기권X   2159
4     3  정기권X   1161
5     4  정기권X    558
6     5  정기권X    658
7     6  정기권X   1888
8     7  정기권X   4966
9     8  정기권X   9802
10    9  정기권X  14085
11   10  정기권X  20984
12   11  정기권X  27324
13   12  정기권X  31387
14   13  정기권X  33771
```

02-1 정식등록과 일일대여 자전거의 working day 여부에 따른 이용시간 추이

시각화

[정식등록과 일일대여 자전거의 렌트 수]



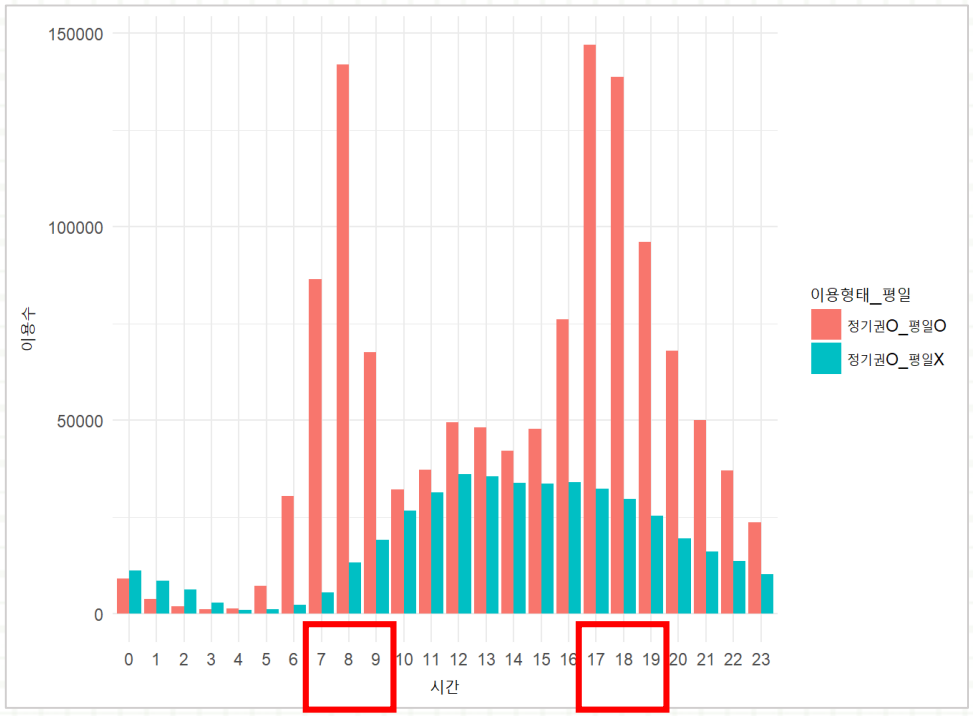
8시, 17~19시 이용자

정기권X와 정기권O의 차이?
출퇴근시간에 차이 커지는 이유는?

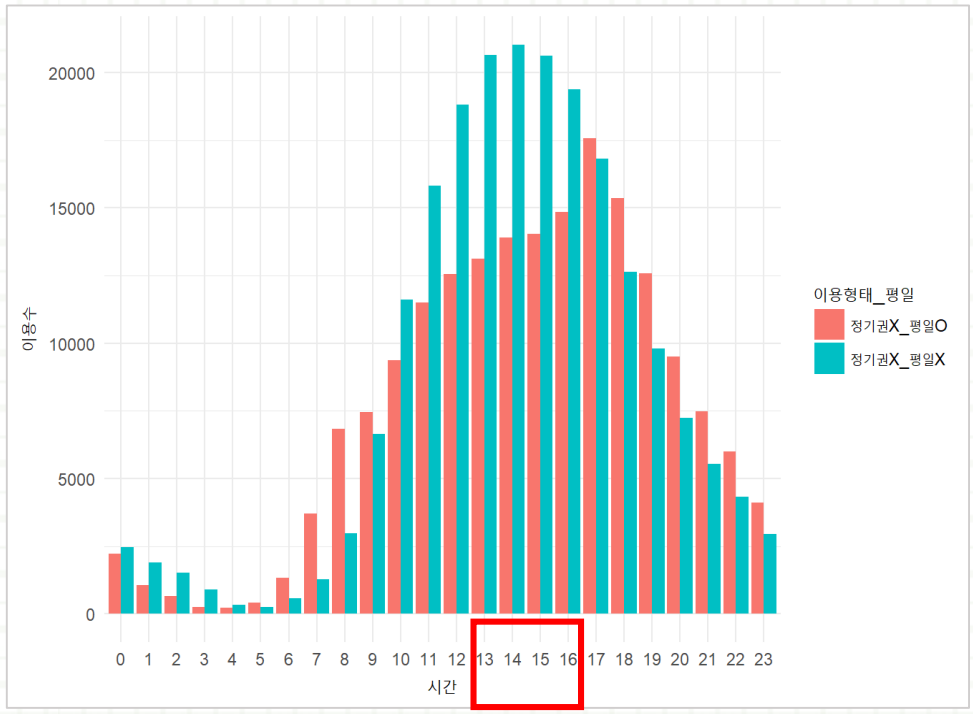
02-1 정식등록과 일일대여 자전거의 working day 여부에 따른 이용시간 추이

시각화

[정식등록 자전거_평일]



[일일대여 자전거 _평일]



출퇴근자 → **정식등록 이용**

02-2 날씨에 따른 정식등록과 일일대여 이용자의 추이

전처리: 날씨에 따른 시간별 이용자 합계

```
weatherbike <- aggregate(train[,c("weather")], by=list(train$Hour, train$weather), "sum")
colnames(weatherbike) <- c('시간', '날씨', '총이용수')
```

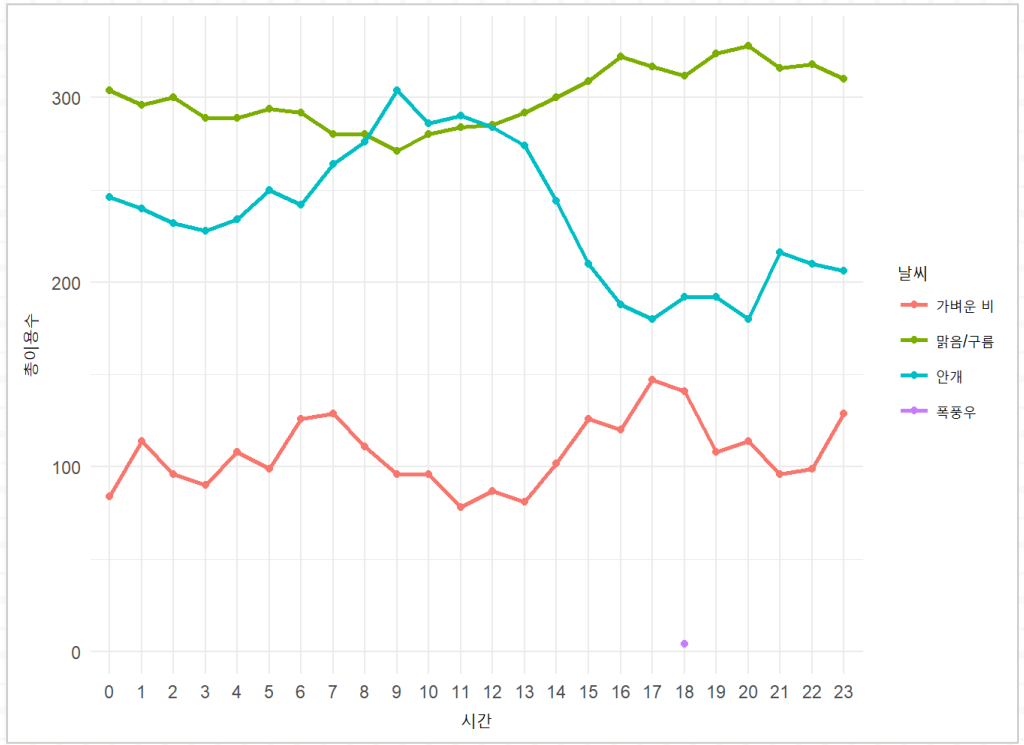
전처리: 날씨의 범주명 변경

```
weatherbike$날씨<-gsub("1","맑음/구름",weatherbike$날씨)
weatherbike$날씨 <-gsub("2","안개",weatherbike$날씨)
weatherbike$날씨 <-gsub("3","가벼운 비",weatherbike$날씨)
weatherbike$날씨 <-gsub("4","폭풍우",weatherbike$날씨)
```

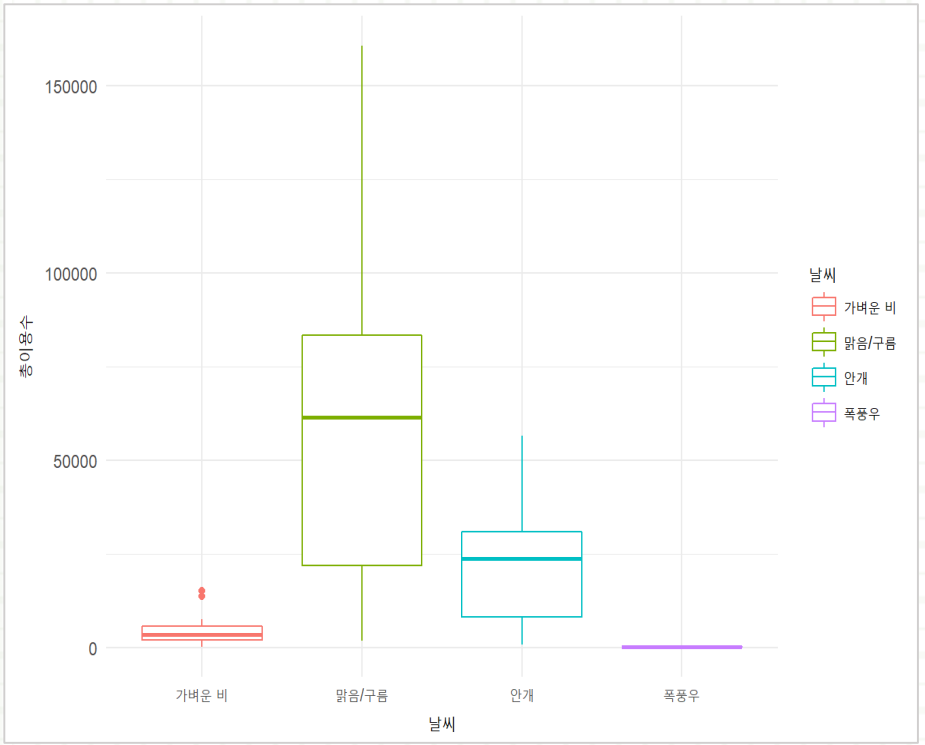
02-2 날씨에 따른 정식등록과 일일대여 이용자의 추이

시각화

[날씨에 따른 총이용자수]



[날씨에 따른 총이용자수_Boxplot]

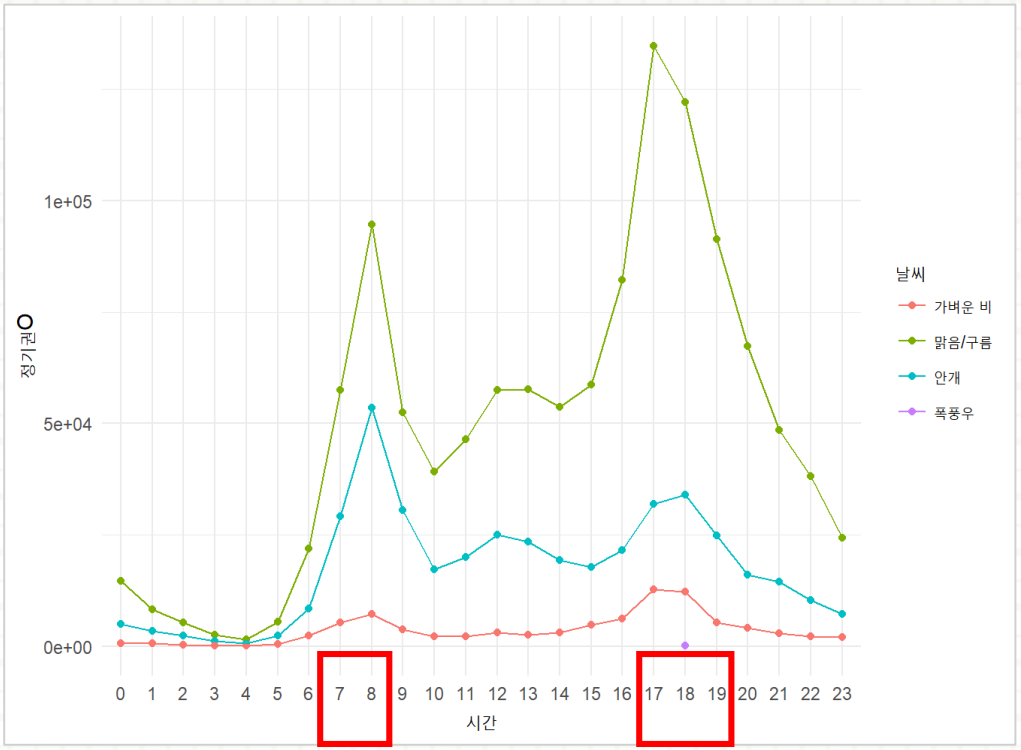


맑음/구름 날씨에 많이 이용

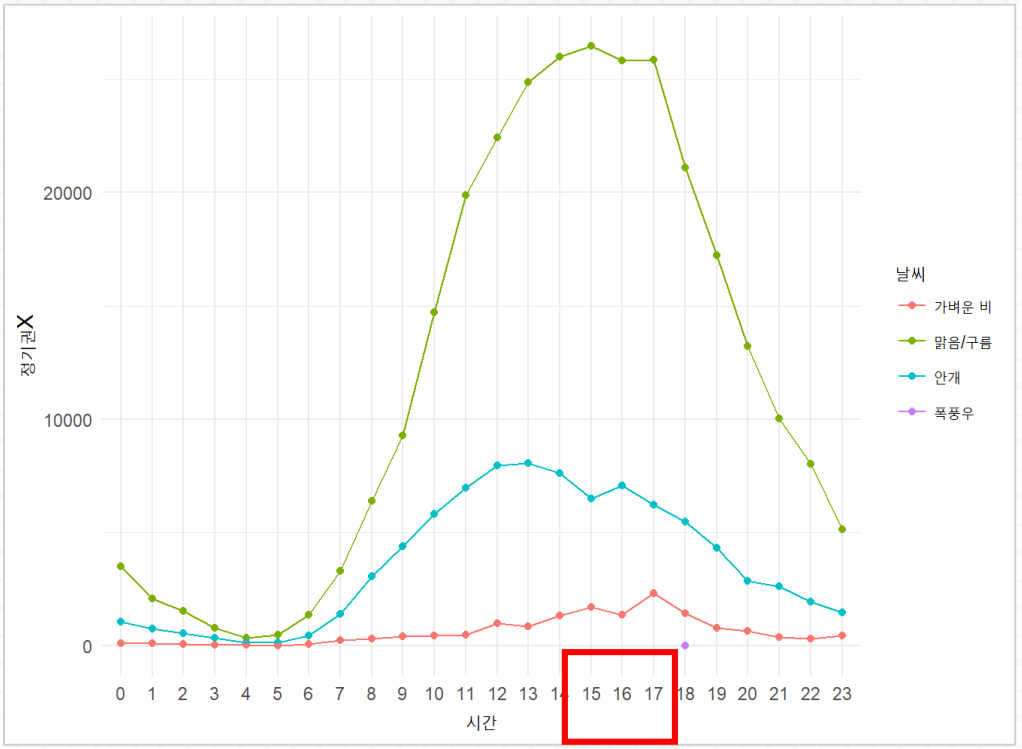
02-2 날씨에 따른 정식등록과 일일대여 이용자의 추이

시각화

[정식등록 자전거]



[일일대여 자전거]

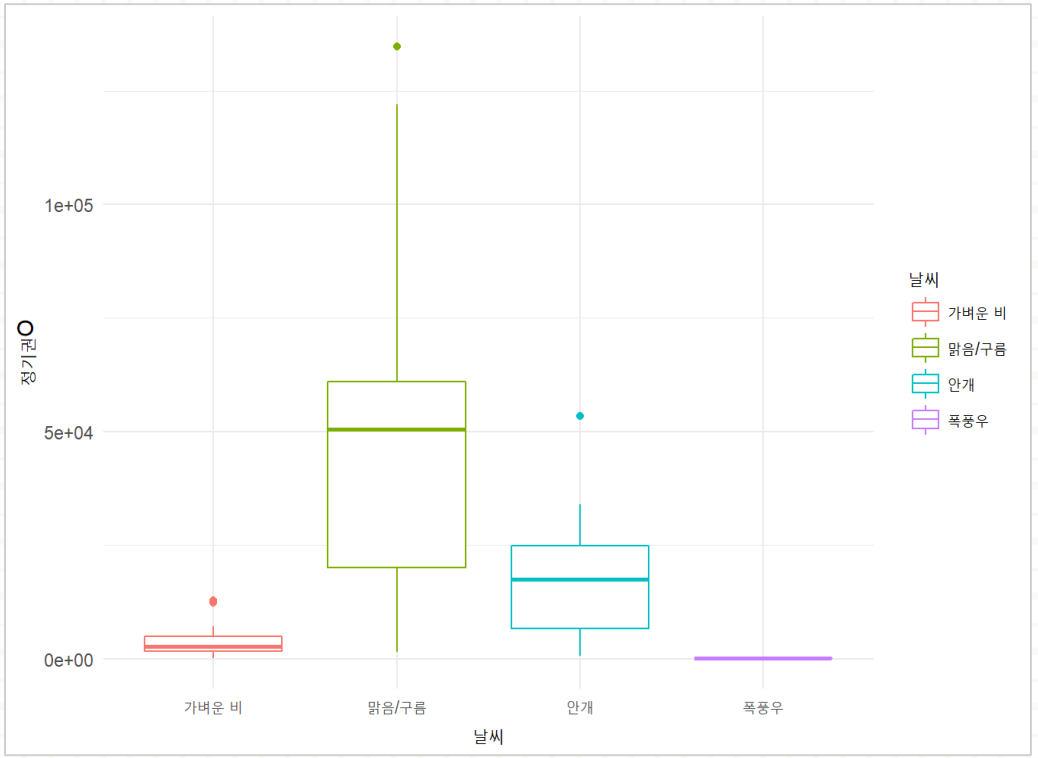


두 종류의 이용자 모두 날씨에 따라서
자전거 이용에 차이가 있는 것을 유추해볼 수 있다.

02-2 날씨에 따른 정식등록과 일일대여 이용자의 추이

시각화

[정식등록 이용자 수_날씨]



[정식등록 이용자 수의 날씨별 anova test]

```
> summary(aov(registered~weather,train))
              Df    Sum Sq Mean Sq F value Pr(>F)
weather        1   2968711 2968711   131.7 <2e-16 ***
Residuals  10884 245348503   22542
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

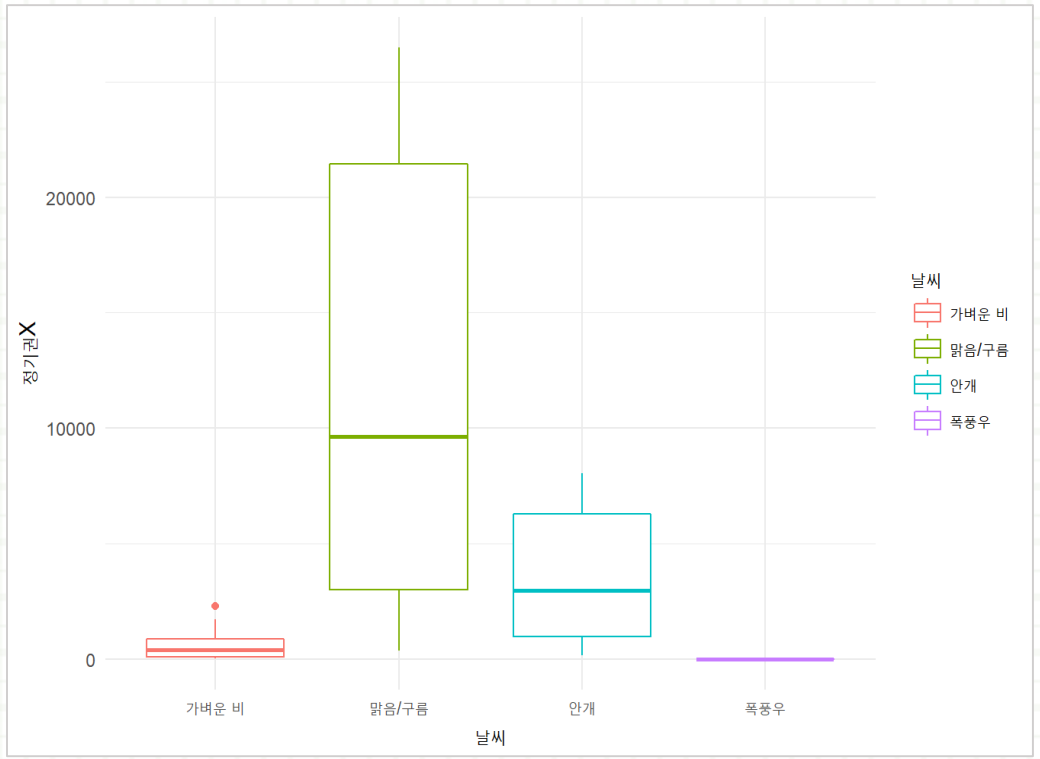
➔ 유의수준 0.05 하에서 유의

(정기권을 가진 사람들의 날씨에 따른 이용추이는 차이가 있다.)

02-2 날씨에 따른 정식등록과 일일대여 이용자의 추이

시각화

[일일대여 이용자 수_날씨]



[일일대여 이용자 수의 날씨별 anova test]

```
> summary(aov(casual~weather,weatherbike2))
             Df    Sum Sq   Mean Sq F value   Pr(>F)
weather       3 1.722e+09 573942941   16.47 3.58e-08 ***
Residuals    69 2.405e+09  34850995
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

➔ 유의수준 0.05 하에서 유의

(정기권을 가지지 않은 사람들의 날씨에 따른 이용추이는 차이가 있다.)

02-3 계절에 따른 정식등록과 일일대여 이용자의 추이

전처리: 계절에 따른 시간별 이용자 합계

```
seasonbike <- aggregate(train[,c("정기권X", "정기권O")], by=list(train$Hour, train$season), "sum")
colnames(seasonbike) <- c('시간', '날씨', '정기권X','정기권O' )
seasonbike$총이용수<-(seasonbike$정기권X+seasonbike$정기권O)
```

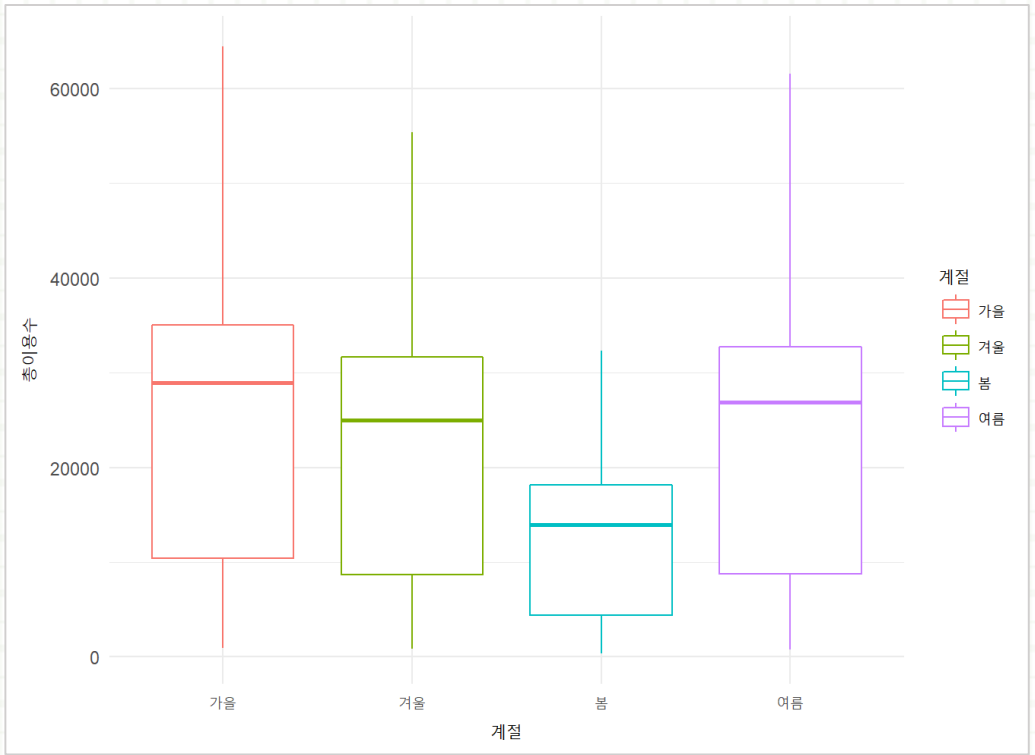
전처리: 계절의 범주명 변경

```
seasonbike$계절<-gsub("1","봄",seasonbike$ 계절)
seasonbike$계절 <-gsub("2","여름",seasonbike$계절)
seasonbike$계절 <-gsub("3","가을",seasonbike$계절)
seasonbike$계절 <-gsub("4","겨울",seasonbike$계절)
```

02-3 계절에 따른 정식등록과 일일대여 이용자의 추이

시각화

[총이용자수_계절]



[총이용자 수의 계절별 anova test]

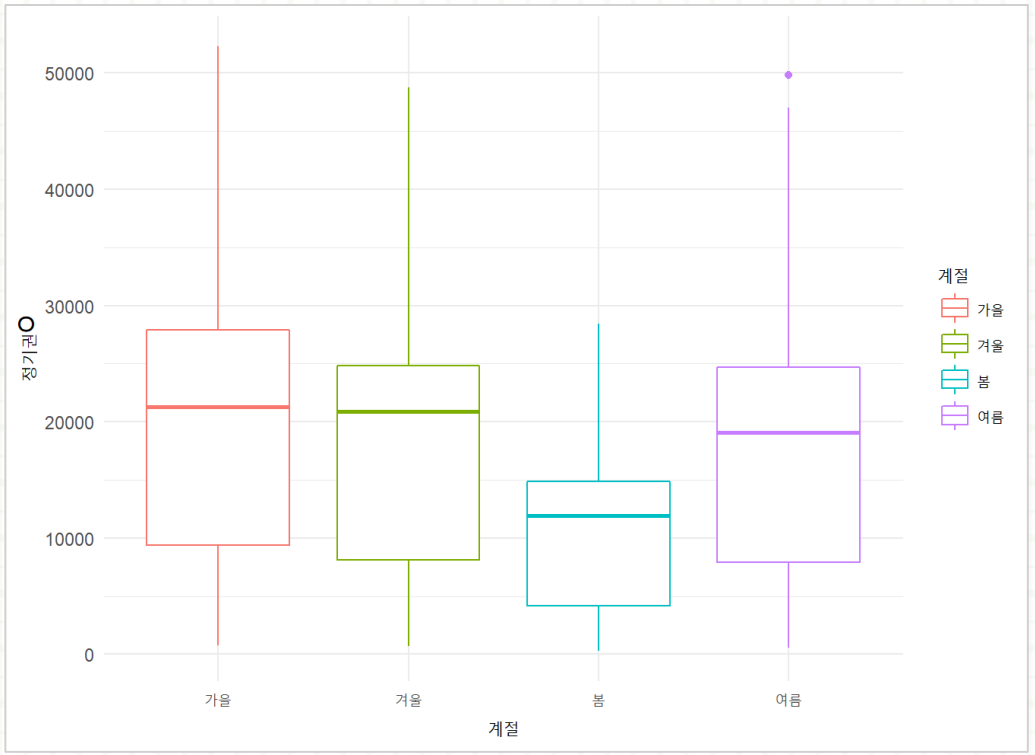
```
> summary(aov(total~season,seasonbike))
             Df    Sum Sq   Mean Sq F value Pr(>F)
season        3 2.619e+09 872902309   3.573  0.017 *
Residuals    92 2.248e+10 244319899
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

➔ 유의수준 0.05 하에서 유의
(계절별로 총 이용자수의 추이에 차이가 있다.)

02-3 계절에 따른 정식등록과 일일대여 이용자의 추이

시각화

[정식 등록 이용자수_계절]



[정식 등록 이용자의 계절별 anova test]

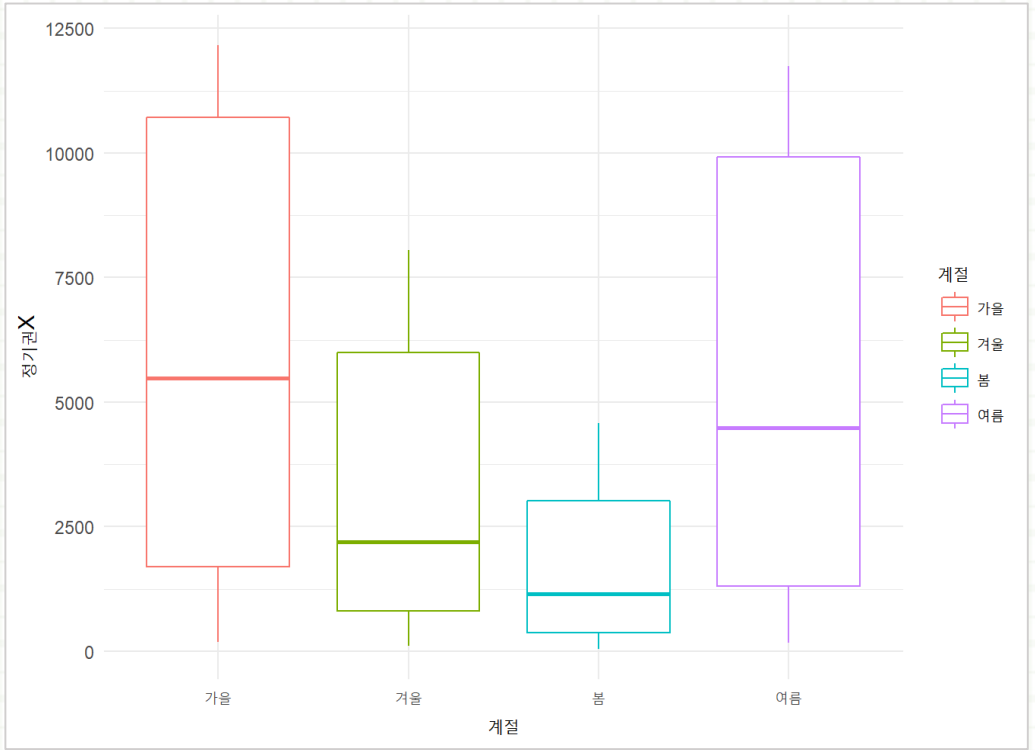
```
> summary(aov(registered~season,seasonbike))
             Df    Sum Sq   Mean Sq F value Pr(>F)
season        3 1.328e+09 442509172   2.576 0.0586 .
Residuals    92 1.581e+10 171805673
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

➔ 유의수준 0.05 하에서 유의하지 않음
(정기이용권을 가진 이용자들은 계절에 따라 이용 추이가 변하지 않는다.)

02-3 계절에 따른 정식등록과 일일대여 이용자의 추이

시각화

[일일대여 이용자수_계절]



[일일대여 이용자의 계절별 anova test]

```
> summary(aov(casual~season,seasonbike))
             Df    Sum Sq  Mean Sq F value   Pr(>F)
season        3 2.741e+08 91355919   7.445 0.000162 ***
Residuals    92 1.129e+09 12270218
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

➔ 유의수준 0.05 하에서 유의
(정기권을 가지지 않은 이용자들의 자전거 이용 추이는 계절에 따라서 차이가 있다.)

02-4 계절간 총이용자수의 차이 분석

전처리: datatime에서 월, 요일, 시간 추출 및 데이터형 변환

```
month <-as.integer(format(as.POSIXlt(train.new$datetime), format = "%m"))
weekday <- as.integer(format(as.POSIXlt(train.new$datetime), format = "%u"))
hour <- as.integer(format(as.POSIXlt(train.new$datetime), format = "%H"))
train <- data.frame(train$season, month, weekday, hour, as.factor(train$workingday),
                    as.factor(train$holiday), as.factor(train$weather), train$temp, train$hum,
                    train$windspeed, train$count)
names(train) <- c("계절", "달", "평일", "시간", "평일여부", "공휴일여부", "날씨", "온도", "습도", "풍속", "이용수")
```

전처리: 풍속의 결측값 제거

```
train <- train[which(train$풍속 != 0.0000),]
```

02-4 계절간 총이용자수의 차이 분석

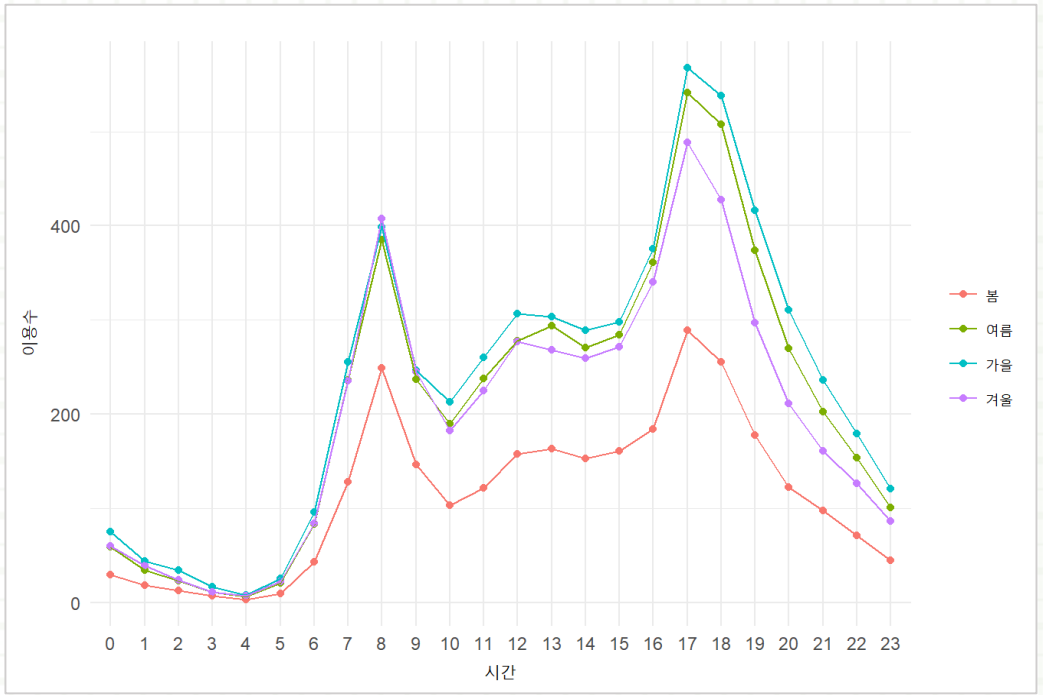
전처리: 계절, 요일, 날씨에 따른 시간별 합계의 평균

```
season.summary <- ddply(train.select,.(계절,시간),
                        summarise, 이용수 = mean(이용수))

day.summary <- ddply(train.select,.(평일=as.factor(평일),hour=as.factor(시간)),
                     summarise, count = mean(이용수))

weather.prob <- ddply(train.select,.(계절 = as.factor(계절), 시간 = as.factor(시간)),
                      summarise, 좋음 = mean(날씨 == "1"),
                      보통 = mean(날씨 == "2"),
                      나쁨 = mean(날씨 == "3"),
                      매우나쁨 = mean(날씨 == "4"))

weather.summary <- ddply(train.select,.(날씨,시간),
                         summarise, 이용수 = mean(이용수))
```

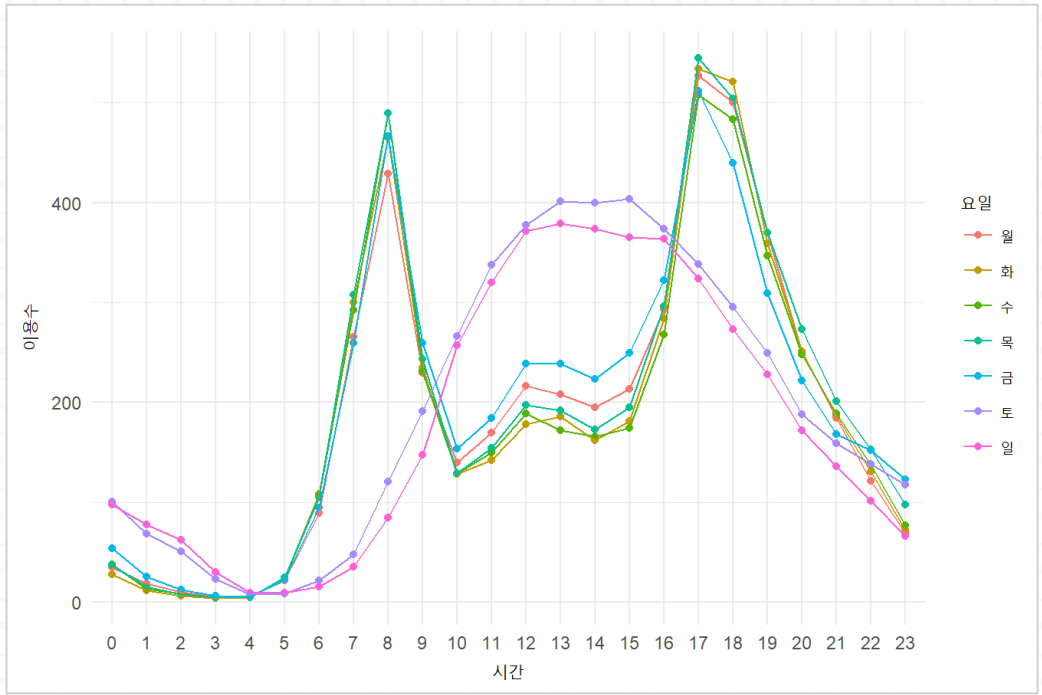
02-4 계절간 총이용자수의 차이 분석

시각화

[계절에 따른 총이용자수]



[요일에 따른 총이용자수]

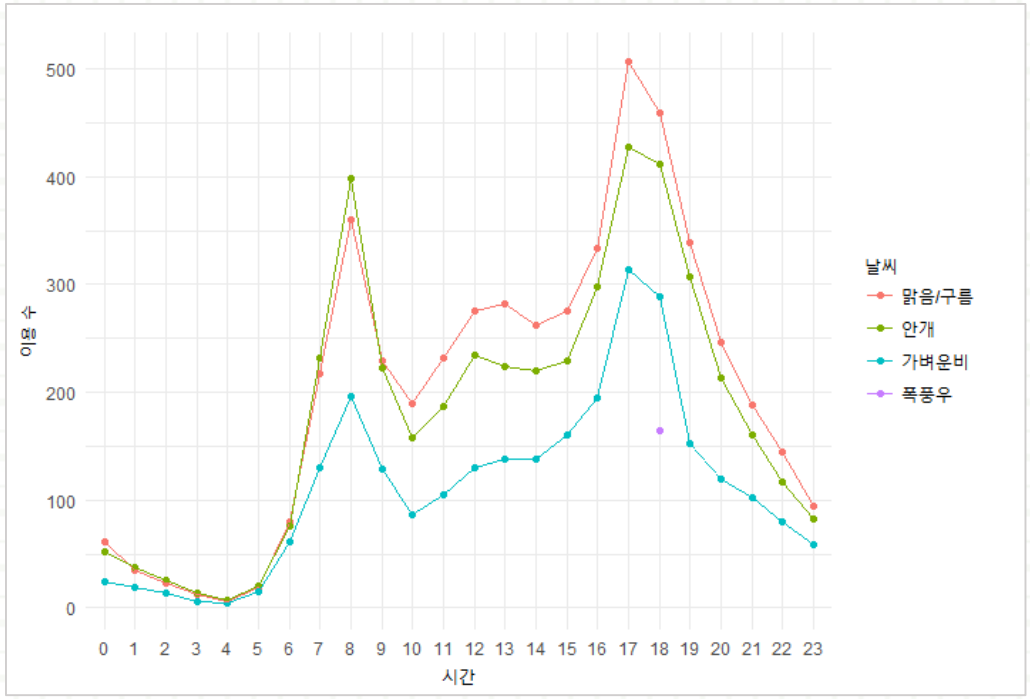


의문? 봄보다 겨울에 이용자수가 더 많다?

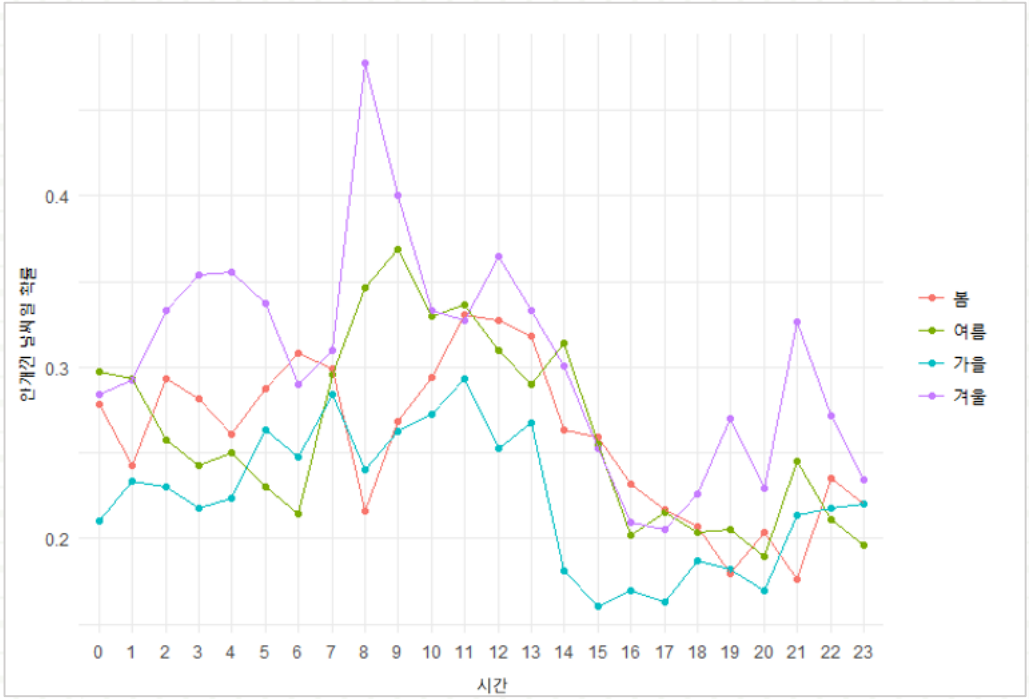
02-4 계절간 총이용자수의 차이 분석

시각화

[날씨별 이용자수]



[안개낀 날씨일 확률]



오전시간에는 안개 낀 날씨에서 이용자가 많으며, 아침 겨울에 안개 낄 확률이 높다.

02-5 습도에 따른 시간대별 자전거 대여건수의 비율 차이

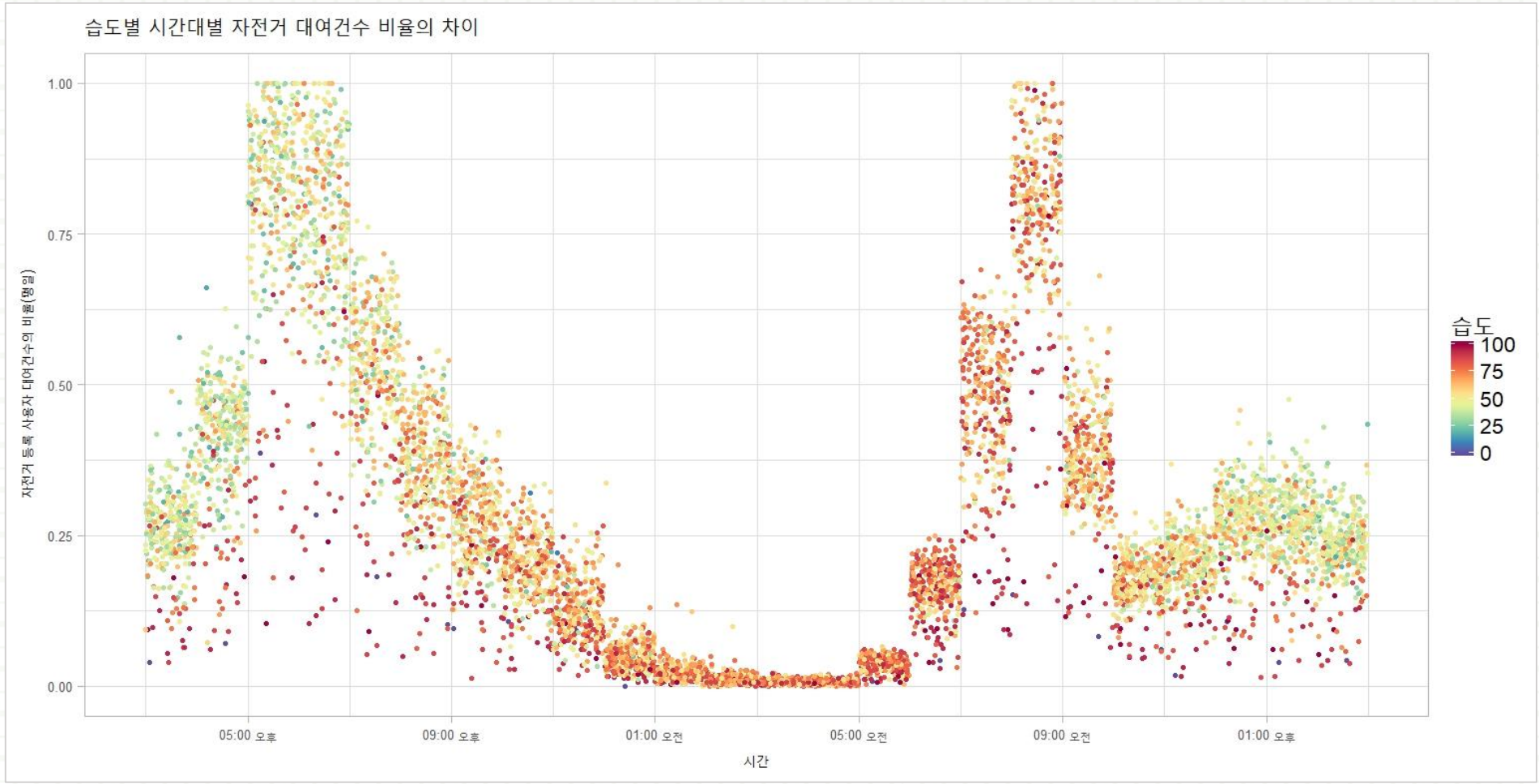
전처리

비율: month_count를 month_count의 max로 나눈 값으로,
즉 month_count를 스케일링 시킨 값

```
count_scale <- tapply(train[train$workingday ==1,]$registered, as.factor(train[train$workingday ==1,]$month),
                      FUN = function(x) x/max(x))
```

02-5 습도에 따른 시간대별 자전거 대여건수의 비율 차이

시각화



분 석

03 분석

전처리: 데이터형 변환

```
train$season <- as.factor(train$season)
train$holiday <- as.factor(train$holiday)
train$workingday <- as.factor(train$workingday)
train$weather <- as.factor(train$weather)
train$datetime <-as.POSIXct(train$datetime, format="%Y-%m-%d %H:%M:%S")
train$day <-  strftime(train$datetime, '%u') # 요일
train$day <- as.factor(train$day)
test$day <-  strftime(test$datetime, '%u')
test$day <- as.factor(test$day)
train$hour <- substring(train$datetime, 12,13)
train$hour <- as.factor(train$hour)
test$hour <- substring(test$datetime, 12,13)
test$hour <- as.factor(test$hour)
```

03 분석

전처리: 의미 없는 변수 제거

```
train=subset(train, select=-c(datetime,casual,registered))
```

(casual+registered=count ➔ count만 남기고 제거)

(datetime은 범주형 연속형 값 ➔ 제거)

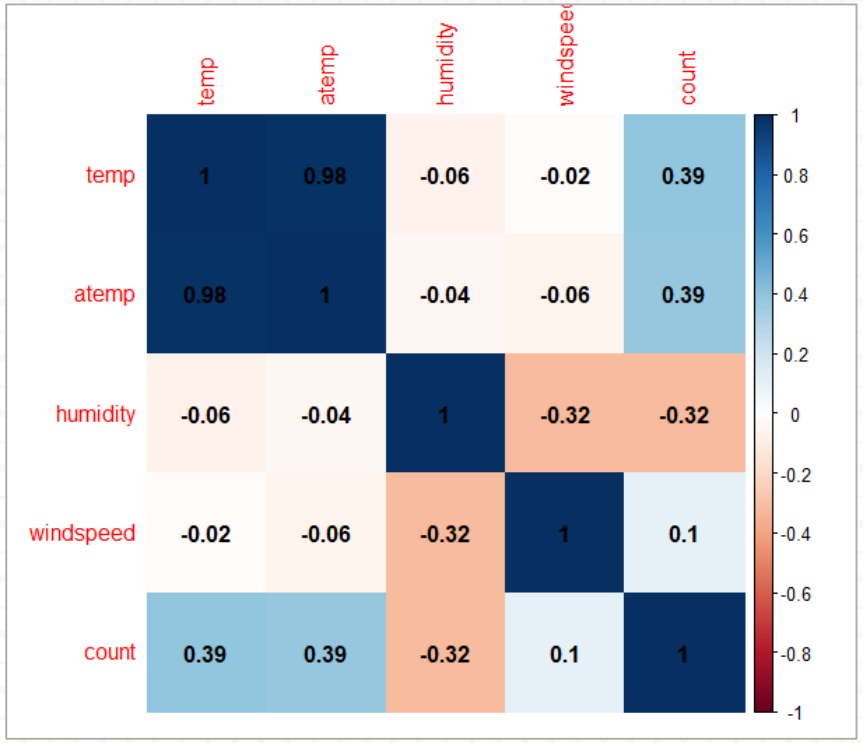
```
train_subset=subset(train, select = c(temp,atemp,humidity,windspeed,count))
```

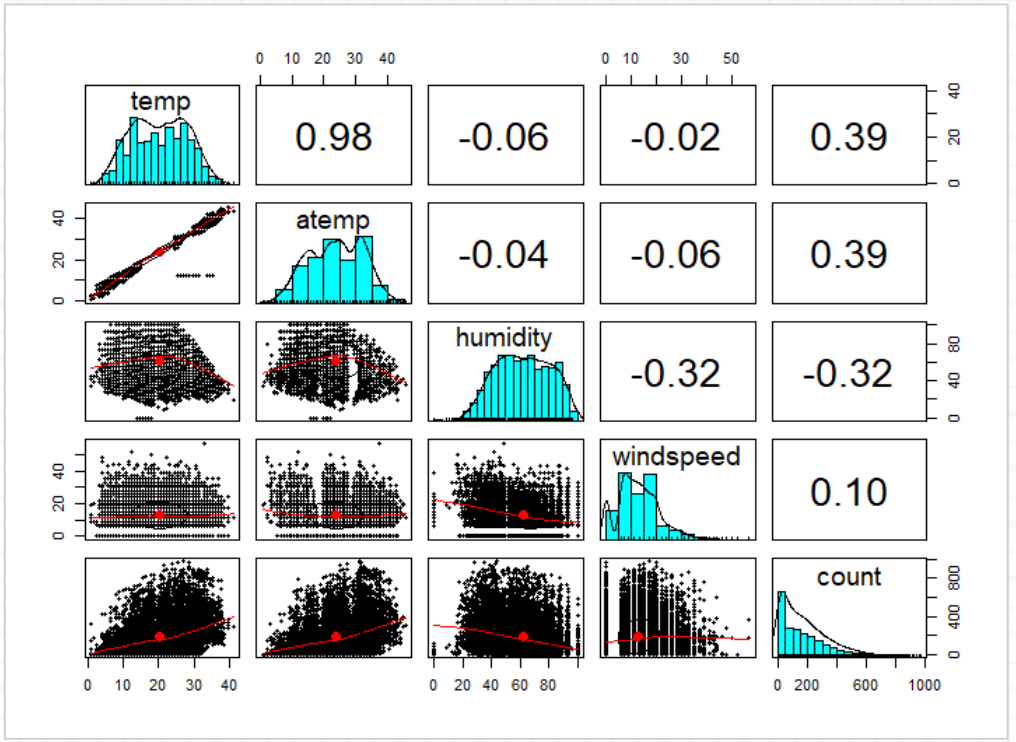
(기상변수 간의 상관관계 & 기상변수와 count의 상관관계를 파악하기 위해 해당 변수만 남김)

```
train_subset$humidity <- as.numeric(train_subset$humidity)
train_subset$count <- as.numeric(train_subset$count)
```

03 분석

상관관계 분석



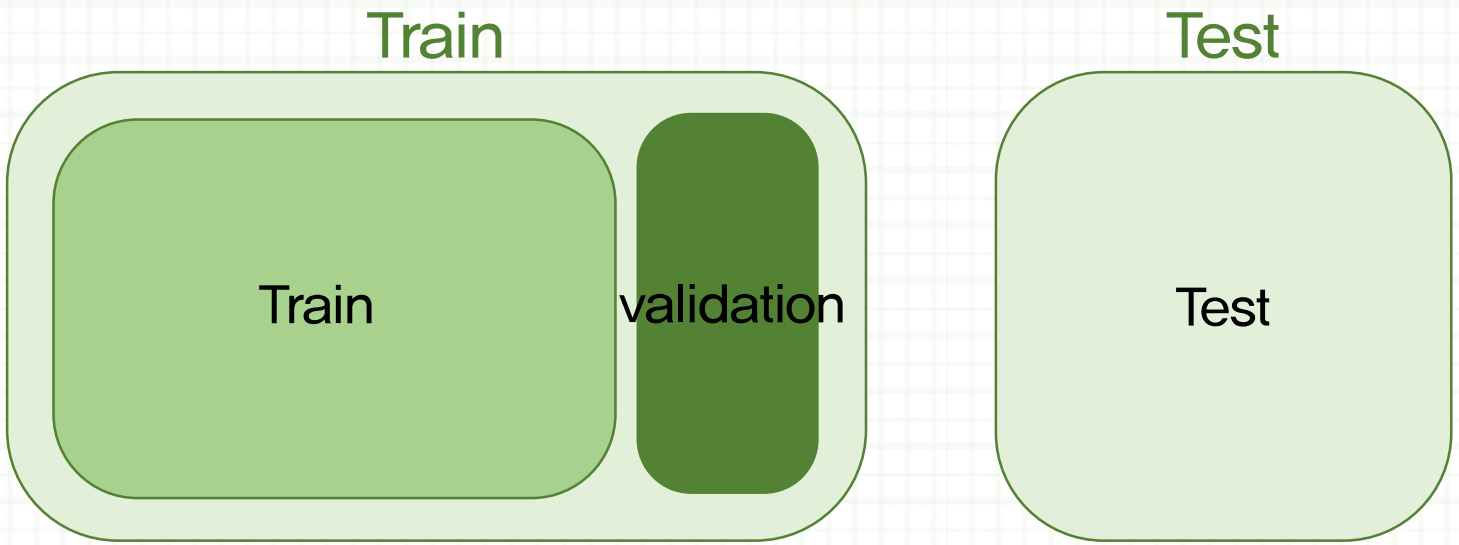


temp와 atemp가 높은 상관관계를 보이며, 총이용자 수는 기온과 약간의 상관관계를 가진다

03 분석

데이터 분할

```
split <- sample.split(train$count, SplitRatio = 0.7)
training <- subset(train, split == TRUE)
validation <- subset(train, split == FALSE
```



03 분석

선형회귀

bikerent <- lm(count~., data = training)

summary(bikerent)

```
Call:
lm(formula = count ~ ., data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-354.56  -61.80  -10.45   52.04  504.79

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -42.93619   10.20862  -4.206 2.63e-05 ***
season2       35.06040    4.72214   7.425 1.25e-13 ***
season3       17.29315    6.05688   2.855 0.004314 **
season4       67.27229    3.90552  17.225  < 2e-16 ***
holiday1       0.06322    8.02450   0.008 0.993714
workingday1    0.41016    4.89570   0.084 0.933234
weather2      -7.31092    3.12567  -2.339 0.019362 *
weather3     -65.72873    5.37677 -12.225  < 2e-16 ***
weather4    -125.12710  111.21701  -1.125 0.260594
temp           4.37802    1.02809   4.258 2.08e-05 ***
atemp          2.28267    0.90191   2.531 0.011396 *
humidity      -0.89676    0.08995  -9.969  < 2e-16 ***
windspeed     -0.66419    0.17377  -3.822 0.000133 ***
day2           8.13642    4.94660   1.645 0.100042
day3          13.18939    4.91352   2.684 0.007284 **
day4           9.39536    4.96889   1.891 0.058684 .
day5          12.38395    4.90561   2.524 0.011608 *
day6          21.84801    4.70313   4.645 3.45e-06 ***
day7                NA         NA      NA       NA
hour01       -17.89677    8.68384  -2.061 0.039344 *
hour02       -26.94005    8.72679  -3.087 0.002029 **
hour03       -36.37484    8.82883  -4.120 3.83e-05 ***
hour04       -39.61364    8.74331  -4.531 5.97e-06 ***
hour05       -23.46618    8.80467  -2.665 0.007711 **
hour06        37.70131    8.72385   4.322 1.57e-05 ***
hour07       179.11468    8.87441  20.183  < 2e-16 ***
hour08       309.75539    8.67838  35.693  < 2e-16 ***
hour09       163.85412    8.69366  18.848  < 2e-16 ***
hour10       103.71612    8.74406  11.861  < 2e-16 ***
hour11       134.78740    8.81449  15.292  < 2e-16 ***
hour12       171.07786    8.81444  19.409  < 2e-16 ***
hour13       164.71838    8.99613  18.310  < 2e-16 ***
hour14       141.95734    8.93442  15.889  < 2e-16 ***
hour15       163.22441    8.97690  18.183  < 2e-16 ***
hour16       222.77200    8.81457  25.273  < 2e-16 ***
hour17       394.13986    8.98938  43.845  < 2e-16 ***
hour18       350.91925    8.95186  39.201  < 2e-16 ***
hour19       232.41356    8.73520  26.607  < 2e-16 ***
hour20       156.37777    8.68308  18.009  < 2e-16 ***
hour21       104.30546    8.63787  12.075  < 2e-16 ***
hour22        69.04542    8.68254   7.952 2.10e-15 ***
hour23        29.22411    8.82148   3.313 0.000928 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 110.9 on 7596 degrees of freedom
Multiple R-squared:  0.635,     Adjusted R-squared:  0.6331
F-statistic: 330.4 on 40 and 7596 DF,  p-value: < 2.2e-16
```

03 분석

변수선택 (혼합선택법 사용)

```
selection<-stepAIC(bikerent, direction="both")
summary(selection)
formula(selection)
```

Step:
AIC = 71955.54

변수선택결과:
count ~ season + weather + temp + atemp + humidity + windspeed + day + hour

03 분석

검증셋 예측

```
> validaion_rmse<-rmse(validation$count,predict_validation)
> print(validaion_rmse)
[1] 108.0763
```

RMSE: 108.0763

실제값과 검증데이터 예측값 비교

[실제값 summary]

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.0	41.0	144.0	188.3	281.0	884.0

[예측값 summary]

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
−160.32	76.23	192.82	193.94	297.35	630.21

(음수값 처리 ➔ Log화)

03 분석

로그화 시킨 후 선형회귀

```
log=lm(log(count)~., data = training)
logselection <- stepAIC(log, direction="both") # 위와 동일하게 진행
predict_validation_log <- predict(logselection,newdata=validation)
```

Step:

AIC = -6097.44

변수선택결과:

```
log(count) ~ season + weather + temp + atemp + humidity + windspeed +
             day + hour
```

03 분석

지수함수를 이용해 log → non-log값으로 변경

```
predict_validation_nonlog <- exp(predict_validation_log)
summary(predict_validation_nonlog)
```

[예측값 summary]

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.388	46.555	143.113	173.325	262.594	1058.837

(음수값 없어진 것 확인)

03 분석

RMSLE

: 과대평가 된 항목보다는 과소평가 된 항목에 페널티를 준다(RMSE와의 차이점).
오차(Error)를 제곱(Square)해서 평균(Mean)한 값의 제곱근(Root)으로 값이 작을수록 정밀도가 높다.
0에 가까운 값이 나올수록 정밀도가 높은 값이다.

```
rmsle(validation$count,predict_validation_nonlog)
```

RMSLE: 0.6264606 ➔ 유의하다

03 분석

Test셋 예측

```
predict_test_log <- predict(logselection,newdata=test)
predict_test_nonlog <- exp(predict_test_log) # log -> non-log
predict <- cbind(as.data.frame(predict_test_nonlog), test$datetime)
colnames(predict) <- c("count", "datetime")
```

```
> head(predict)
      count             datetime
1 21.932944 2011-01-20 00:00:00
2 12.748189 2011-01-20 01:00:00
3  7.499473 2011-01-20 02:00:00
4  4.112942 2011-01-20 03:00:00
5  3.028764 2011-01-20 04:00:00
6  8.357070 2011-01-20 05:00:00
```

03 분석

따릉이 수요 예측 모델 구축

```
# 로그화 시킨후 선형회귀
log=lm(log(count)~., data = training)
logselection <- stepAIC(log, direction="both")

predict_validation_log <- predict(logselection,newdata=validation)

# 지수함수를 이용해 log -> non-log 값으로 변경
predict_validation_nonlog <- exp(predict_validation_log)
summary(predict_validation_nonlog) # 음수값이 존재X

rmsle(validation$count,predict_validation_nonlog)
```

```
> formula(logselection)
log(count) ~ temp + humidity + weather
+ windspeed + atemp + season
```

➔ 기상 변수로만 이루어진 모델 구축

```
>rmsle(validation$count,predict
_validation_nonlog)
[1] 1.198273
```

0.6264606 ➔ 1.200047

03 분석

따릉이 수요 예측

```
# test셋 예측
predict_test_log <- predict(logselection,newdata=korea)
predict_test_nonlog <- exp(predict_test_log) # log -> non-log
predict <- cbind(as.data.frame(kor_date), predict_test_nonlog)
colnames(predict) <- c("datetime","count")
```

```
>rmsle(predict$count,korea2$count)
[1] 3.510937
```

RMSLE: 3.510937

➔ 매우 부정확하게 예측

```
> head(predict) # 예측값
    datetime     count
1 2015-09-18 73.09669
2 2015-09-19 67.81142
3 2015-09-20 84.38746
4 2015-09-21 107.73648
5 2015-09-22 71.76512
6 2015-09-23 85.16080
```

```
➢ head(korea[,1:2]) # 실제값
      datetime count
➢ 1 2015-09-19   724
➢ 2 2015-09-20   907
➢ 3 2015-09-21   553
➢ 4 2015-09-22   796
➢ 5 2015-09-23   804
➢ 6 2015-09-24   914
```

결 론

04 결론

- ✓ 자전거 정기권 등록 이용자는 미등록 이용자에 비해 계절&날씨에 영향을 적게 받는다.
- ✓ 자전거 이용량은 봄보다는 겨울에 많다.
- ✓ 오전시간 자전거 이용량은 맑은 날씨보다 안개 낀 날씨에 보다 많다.
- ✓ 겨울철 오전시간에는 안개가 자주 끼므로 자전거 이용량이 많다.
- ✓ 자전거 이용에 미치는 미국&한국의 기상 특성은 각기 매우 다르다.

감사합니다