[R] R을 통해 시계열 그래프 만들기 (자료 분석을 위한 시각화와 실습 예제) — 나무늘보의 개발 블로그

노트북: blog
만든 날짜: 2020-10-04 오후 7:19
URL: https://continuous-development.tistory.com/47?category=793392

R

[R] R을 통해 시계열 그래프 만들기 (자료 분석을 위한 시각화와 실습 예제)

2020. 7. 30. 22:37 수정 삭제 공개

#시계열 - 시계열(time series) 데이터는 관측치가 시간적 순서를 가진 데이터

여기서는 시간의 흐름에 따라 값이 변화하는 것을 그래프로 만드는 것을 목적으로 한다.
일단 첫 번째 iris는 시계열 데이터는 아닌 것 같다. 행으로 나눠서 행이 변화하는 것에 따라서 그래프를 그린다.

```
> # 시계열(time series) - 시계열(time series) 데이터는 관측치가 시간적 순서를 가진 데이터
> # 변수간의 상관성
>
> # iris 시계열 데이터 만들기
>
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
```

rownames로 행의 값을 추출해낸다. 처음에는 char데이터여서 이것을 타입 변환을 해줘야 한다.

```
> rownames(iris)
  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"  "25"
 [26] "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"  "49"  "50"
 [51] "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"
 [76] "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"  "100"
[101] "101" "102" "103" "104" "105" "106" "107" "108" "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121" "122" "123" "124" "125"
[126] "126" "127" "128" "129" "130" "131" "132" "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144" "145" "146" "147" "148" "149" "150"
> seq<- as.integer(rownames(iris)) # 타입변환
>
> ?cbind
>
> irisDF <-cbind(seq,iris)
>
> irisDF
    seq Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1     1          5.1         3.5          1.4         0.2     setosa
2     2          4.9         3.0          1.4         0.2     setosa
3     3          4.7         3.2          1.3         0.2     setosa
4     4          4.6         3.1          1.5         0.2     setosa
5     5          5.0         3.6          1.4         0.2     setosa
```

타입변환 한 값을 변수에 저장하고 이 변수에 cbind로 기존에 있던 iris로 합쳐준다.
여기서 colsColor 하는 작업은 컬러 값을 주기 위해서 뽑아왔다. 이 부분은 생략해도 된다. 컬러 값을 뽑아오고 그 값들의 속성명을 irisDF의 속성 값으로 한다. 여기서 iris의 속성 값들은 2:5로 우리가 필요한 variable 값이다. 우리는 이 값들을 행의 값에 변화에 따라서 4가지 그래프를 그릴 것이다.

```
>
> #x축은 seq
> #y 축은 -Sepcies
>
> colsColor <- topo.colors(4, alpha = .4) # rgb코드값을 랜덤하게 뽑아온다. 그다음 alpha는 투명도를 나타낸다.
> colsColor
[1] "#4C00FF66" "#00E5FF66" "#00FF4D66" "#FFFF0066"
>
> names(colsColor) <- names(irisDF)[2:5]
>
> irisDF
    seq Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1     1          5.1         3.5          1.4         0.2     setosa
2     2          4.9         3.0          1.4         0.2     setosa
3     3          4.7         3.2          1.3         0.2     setosa
4     4          4.6         3.1          1.5         0.2     setosa
5     5          5.0         3.6          1.4         0.2     setosa
6     6          5.4         3.9          1.7         0.4     setosa
7     7          4.6         3.4          1.4         0.3     setosa
8     8          5.0         3.4          1.5         0.2     setosa
9     9          4.4         2.9          1.4         0.2     setosa
10   10          4.9         3.1          1.5         0.1     setosa
11   11          5.4         3.7          1.5         0.2     setosa
12   12          4.8         3.4          1.6         0.2     setosa
13   13          4.8         3.0          1.4         0.1     setosa
```

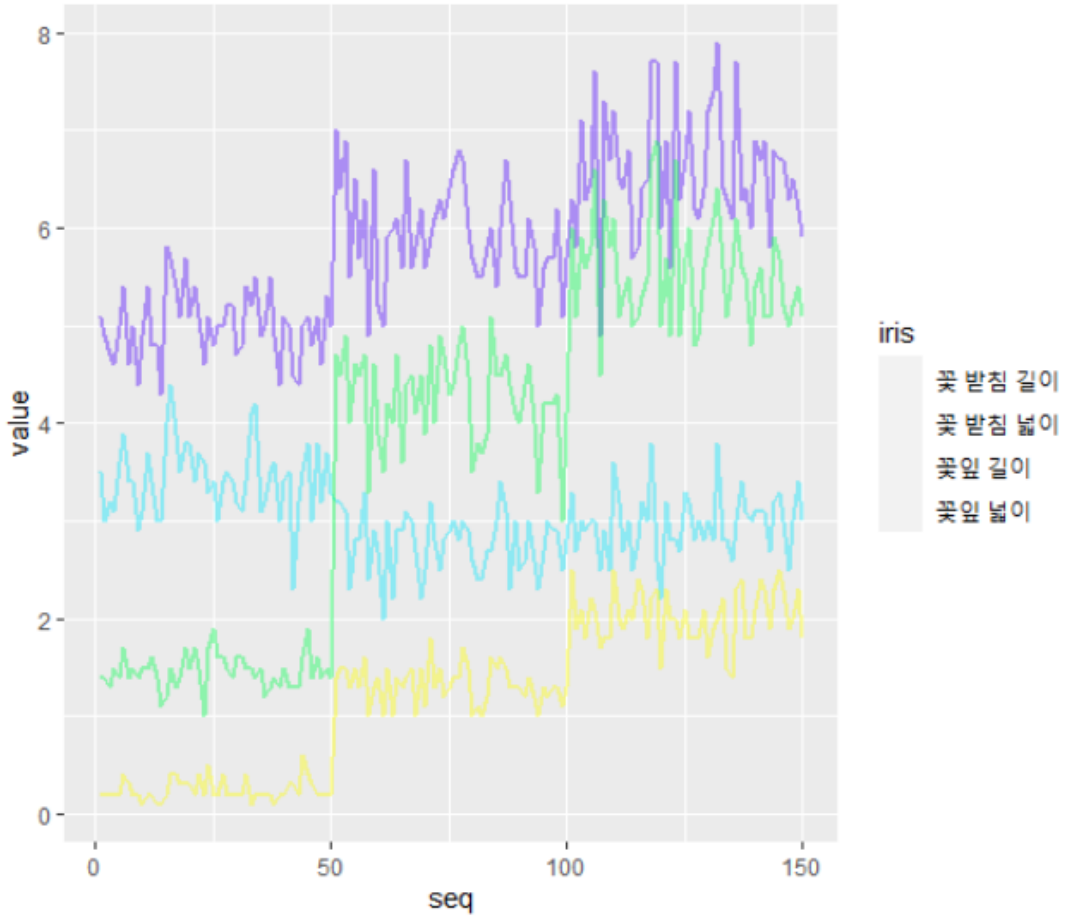
여기서 melt를 사용한다. melt는 가로축의 데이터를 세로축으로 만드는 함수이다. 여기서 가로축으로 된 데이터 Sepal.Length, Sepal.Width, Pet

al.length, Petal.Width를 세로축으로 바꿔주는 작업을 한다. 그때 기준을 seq(행 번호)와 Species을 기준으로 잡는다. 우리가 그릴 그래프는 Species(종)에 따라서 다양한 variable이 바뀌는 것을 보기 위함이기 때문이다.

```
>
> library(reshape2)
>
> # melt 함수를 이용해서 기준 seq, species
> # 나머지 컬럼을 variable 해서 wide → long
>
> # melt 가로로 된걸 세로로 길게 만든다.
> # cast(dcast, acast) - 동일한 결과를 리턴하는데 array 또는 data.frame으로 만드는 함수
>
>
> ?melt
> # melt(데이터를 구분하는 식별자, 측정대상 변수 , 측정치)
>
> iris_melt ← melt(irisDF, id=c("seq","Species"))
> iris_melt
    seq    Species     variable value
1     1     setosa Sepal.Length   5.1
2     2     setosa Sepal.Length   4.9
3     3     setosa Sepal.Length   4.7
4     4     setosa Sepal.Length   4.6
5     5     setosa Sepal.Length   5.0
6     6     setosa Sepal.Length   5.4
7     7     setosa Sepal.Length   4.6
```

데이터를 생성한 후 ggplot에 그래프를 그린다.
seq가 변화함에 따라 종의 변화를 보는 것 이기 때문에 x 축을 seq로 잡고 각각의 variable에 따라 색을 주는 것으로 구분을 하기 때문에 col 값을 variable로 만든다. 그다음 꺾은선그래프를 만들기 위해 geom_line을 사용했다.

```
296  library(ggplot2)
297
298  #시간이 지남에 따라 값이 바뀌는 걸 볼 수있다.
299  g<-ggplot(iris_melt, aes(x=seq,y=value, col=variable))+  #col로 색을 넣는다. 이걸로 묶어서 나오게 한다
300    geom_line(cex=0.8, show.legend = T) #cex 선 두께 legend는 범례이다.
301  g
```



#문자 변수 날짜 변수로 변환(시계열을 위한 준비 작업)

날짜 데이터가 char 형태로 되어 있는 경우가 있다. 이경우에는 날짜 데이터 타입으로 바꿔줘야 시계열 작업이 가능함으로 이런 식으로 바꿔준다.

```
> # 날짜
> # 문자변수를 날짜 변수로 변환
>
> # R의 날짜 데이터 타입 "POSIXct"
> # as.POSIXct()
>
> str_date <-"200730 13:40"
> as.POSIXct(str_date, format="%y%m%d %H:%M")
[1] "2020-07-30 13:40:00 KST"
>
> str_date <-"2020-07-30 13:40:01 PM"
> as.POSIXct(str_date, format = "%Y-%m-%d %H:%M:%S")
[1] "2020-07-30 13:40:01 KST"
>
> str_date <- "07/30/20 13:40:01"
> as.POSIXct(str_date, format = "%m/%d/%y %H:%M:%S")
[1] "2020-07-30 13:40:01 KST"
> |
```

시계열 예제

데이터를 뽑아온 후에 as.POSIXcs 를 통해서 날짜 데이터 타입으로 바꾼다.

```
353
354  # 시계열 - 코스피 예제
355  cospi_time <- read.csv(file.choose()) #데이터를 읽고
356
357  # 여기서 날짜를 시간으로 사용해야 되기 때문에 문자로 되어 있는 날짜를 날짜형식으로 바꾼다.
358  cospi_time$Date<-as.POSIXct(cospi_time$Date, format = "%Y-%m-%d")
359  cospi_time
```

```
> cospi_time
          Date    Open    High     Low   Close Volume
1   2016-02-26 1180000 1187000 1172000 1172000 176906
2   2016-02-25 1172000 1187000 1172000 1179000 128321
3   2016-02-24 1178000 1179000 1161000 1172000 140407
4   2016-02-23 1179000 1189000 1173000 1181000 147578
5   2016-02-22 1190000 1192000 1166000 1175000 174075
6   2016-02-19 1187000 1195000 1174000 1190000 175889
7   2016-02-18 1203000 1203000 1178000 1187000 211795
8   2016-02-17 1179000 1201000 1169000 1185000 245929
9   2016-02-16 1158000 1179000 1157000 1168000 179087
10  2016-02-15 1154000 1160000 1144000 1154000 182471
11  2016-02-12 1130000 1151000 1122000 1130000 254115
12  2016-02-11 1118000 1137000 1118000 1130000 304899
```

여기서 melt를 통해서 data와 볼륨에 따른 variable 값을 가지게 만든다. 우리가 찾을 것은 시간의 변화에 따른 variable(open,high,low,close)를 구하는 것이다.

```
> # 여기서 variable을 만들어야된다. melt로 data와 volume을 기준으로 잡고 나머지 4가지 값인  open, high, low, close가 필요하다.
> # 그래서 melt로 id 와 volume 을 제외한 나머지 가로값을 세로값으로 묶기위해 melt를
> # 사용하고 나오는 결과값을 보면 date에 따른 open, high, low, close 이 생긴다.(Volume을 같이 묶은 이유는
> # 여기서는 나머지 4개의 값을 variable로 쓰기위해서이다.)
> cospi_time_melt ← melt(cospi_time, id=c("Date","Volume"))
> cospi_time_melt
         Date Volume variable   value
1  2016-02-26 176906     Open 1180000
2  2016-02-25 128321     Open 1172000
3  2016-02-24 140407     Open 1178000
4  2016-02-23 147578     Open 1179000
5  2016-02-22 174075     Open 1190000
6  2016-02-19 175889     Open 1187000
7  2016-02-18 211795     Open 1203000
8  2016-02-17 245929     Open 1179000
```

```
367
368 #이걸 여기에 넣어주는데 x에 날짜를 넣고 y축에는 value를 넣어준다. 모든값에 대한 value 이다 그다음 색으로 나머지를 묶어
369 #표현해준다.
370 ggplot(cospi_time_melt, aes(x=Date,y=value,col=variable   ))+
371   geom_line()
372
```

```
368 # 이걸 여기에 넣어주는데 x에는 시계열 이니 시간의 흐름에 따라 변화를 보기 위해
369 # 날짜를 넣고 y축에는 variable에 대한 값을 보기 위해서 value를 넣어준다.
370 # 그다음 col 를 variable 을 사용함으로써 group으로 묶어주는 효과와 동시에 같은 값끼리 같은 색을 나타나게
371 # 표현해준다.
372 ggplot(cospi_time_melt, aes(x=Date,y=value,col=variable   ))+
373   geom_line()
374
375
376
```

그 결과는 이렇게 나온다.

캐글 실습 예제

```r
# 1.
# 데이터 내에 결측치 여부를 확인한다.
# NA값이 310681개 있는 것을 확인할 수 있다.

sum(is.na(trains))
str(trains)
```

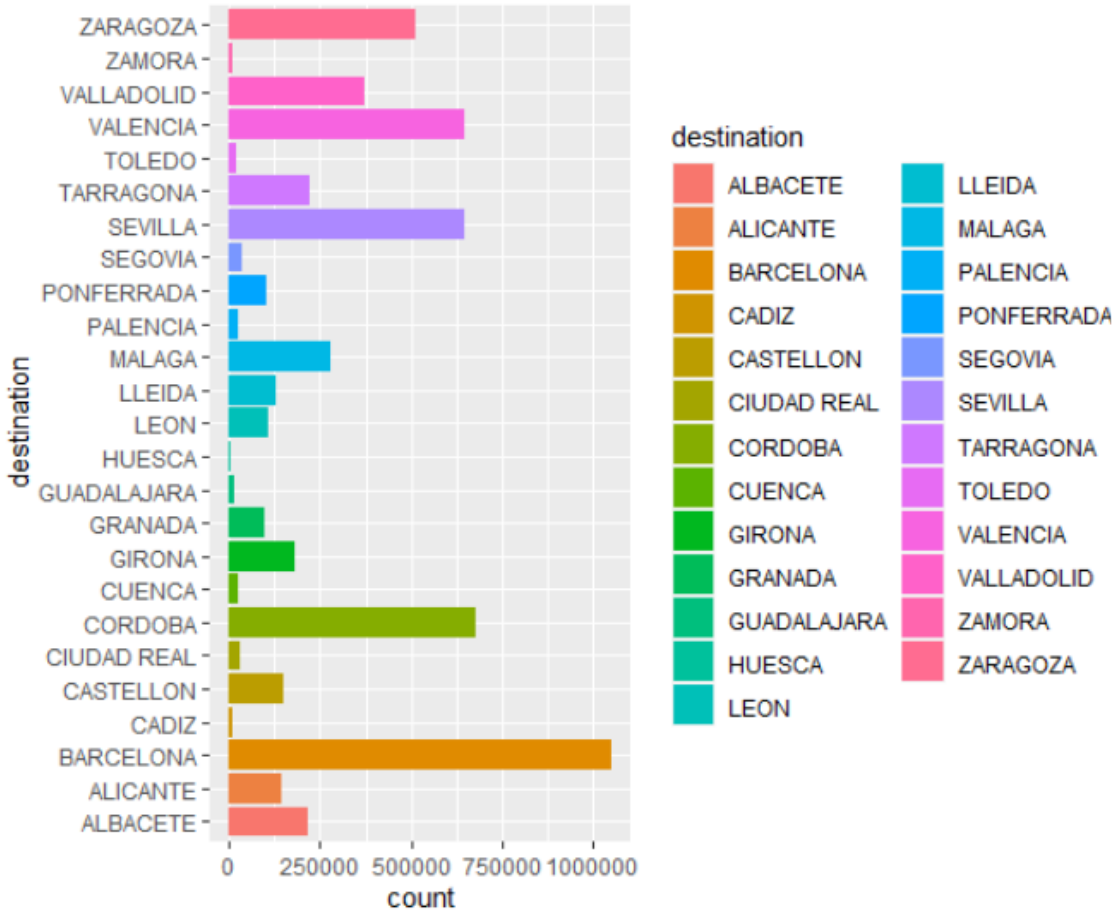
```r
# 2.
# filter와 !is.na함수를 통해 결측치를 모두 제거했다.

trainsNa<-na.omit(trains)
sum(is.na(trainsNa))
```

```r
# 3.
# 마드리드 출발
# 마드리드에서 출발하는 열차 데이터만을 떼어내 madrid_origin이라는 변수로 저장하고
# 우선, 마드리드에서 출발하는 열차 데이터만을 이용해 비교해보기로 한다.

head(trainsNa)
str(trainsNa)
madrid_origin <- subset(trainsNa, origin=='MADRID')
madrid_origin$origin
```

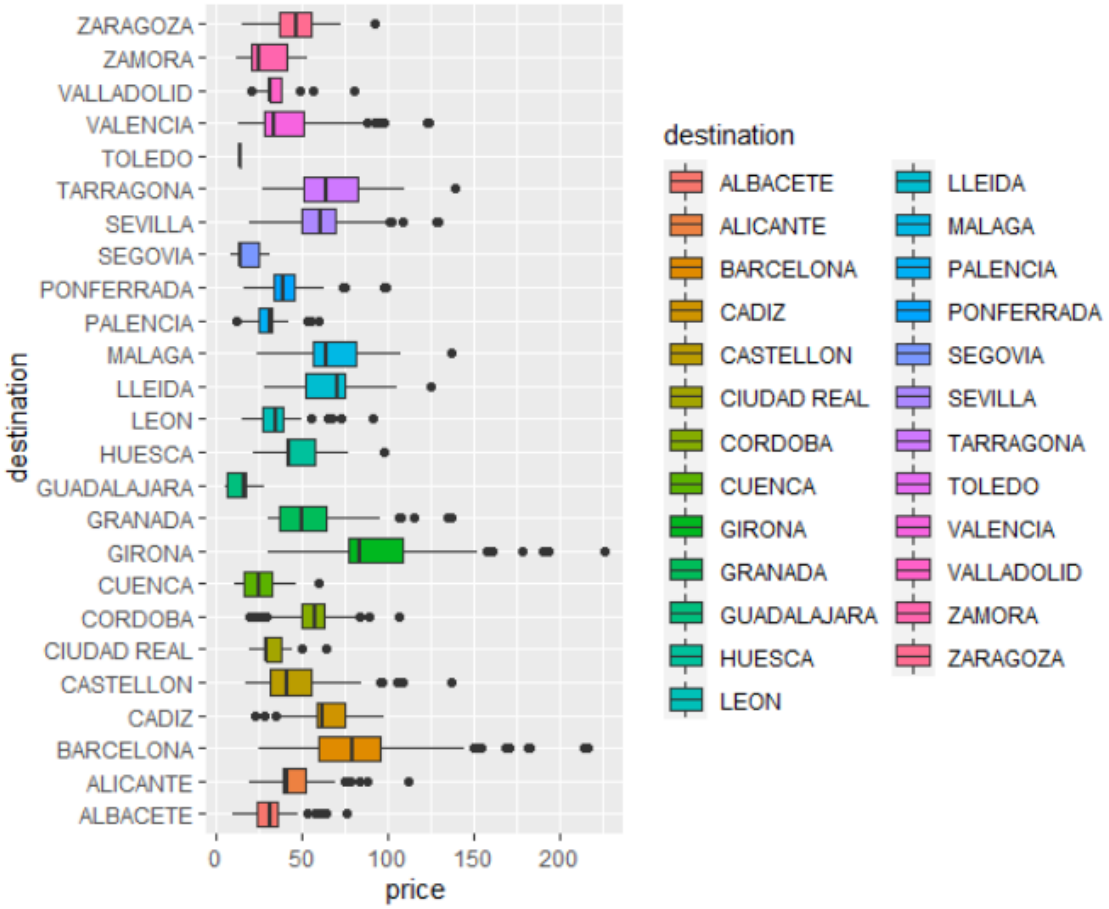
```r
# 4.
# summary함수를 통해 일반적 데이터 정보를 다시 확인한다.
summary(madrid_origin)
```

```
> summary(madrid_origin)
 insert_date          origin          destination         start_date          end_date          train_type          price          train_class
Length:5699800     Length:5699800     Length:5699800     Length:5699800     Length:5699800     Length:5699800     Min.   :  5.65   Length:5699800
Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   1st Qu.: 37.80   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median : 53.60   Mode  :character
                                                                                                                 Mean   : 55.33
                                                                                                                 3rd Qu.: 69.00
                                                                                                                 Max.   :226.40

     fare           price_tree           batch                id
 Length:5699800     Length:5699800     Length:5699800     Min.   :      1
 Class :character   Class :character   Class :character   1st Qu.: 2934984
 Mode  :character   Mode  :character   Mode  :character   Median : 5976612
                                                         Mean   : 6069570
                                                         3rd Qu.: 8977866
                                                         Max.   :19422194
```

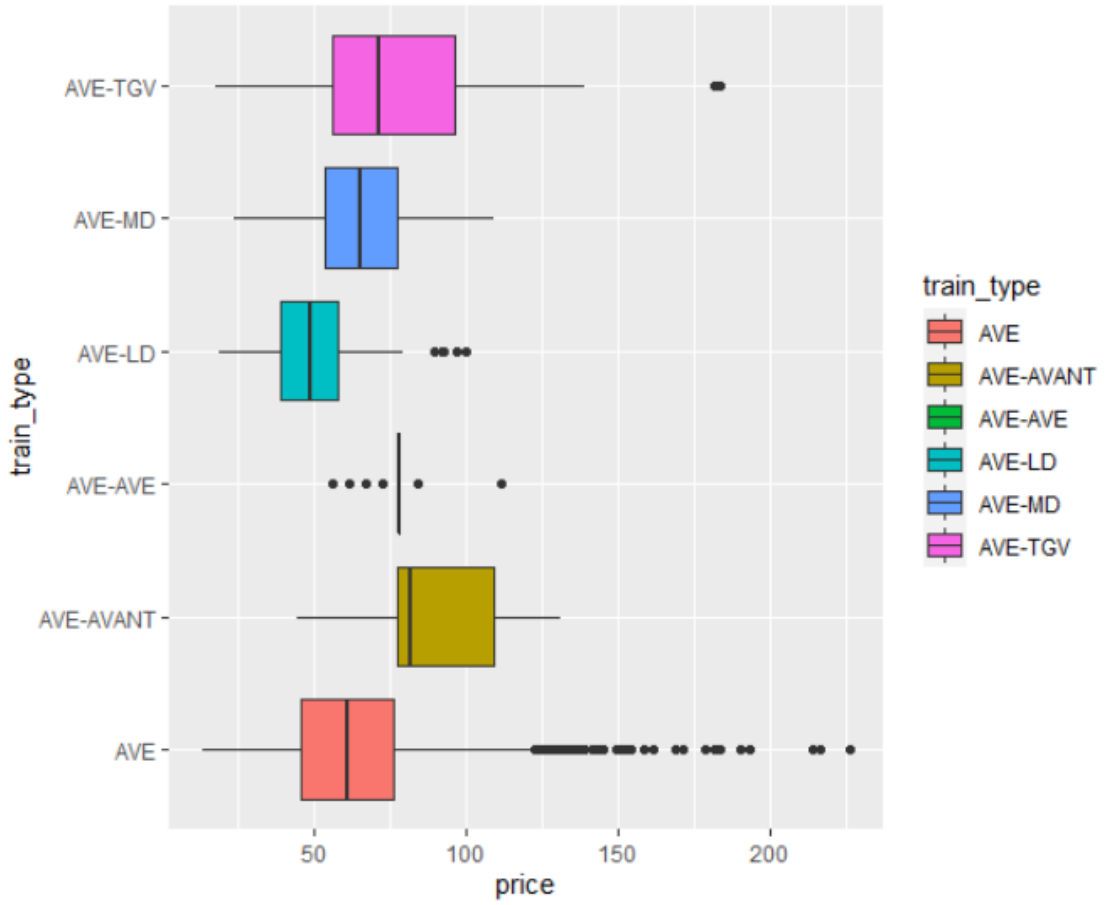
```r
# 5.
# 마드리드 출발 열차의 빈도 수
# 마드리드를 출발하는 기차의 도착 도시별 운행빈도 수를 바형태로 나타내보자
ggplot(madrid_origin,aes(x=destination,fill=destination))+
  geom_bar()+
  coord_flip()
```

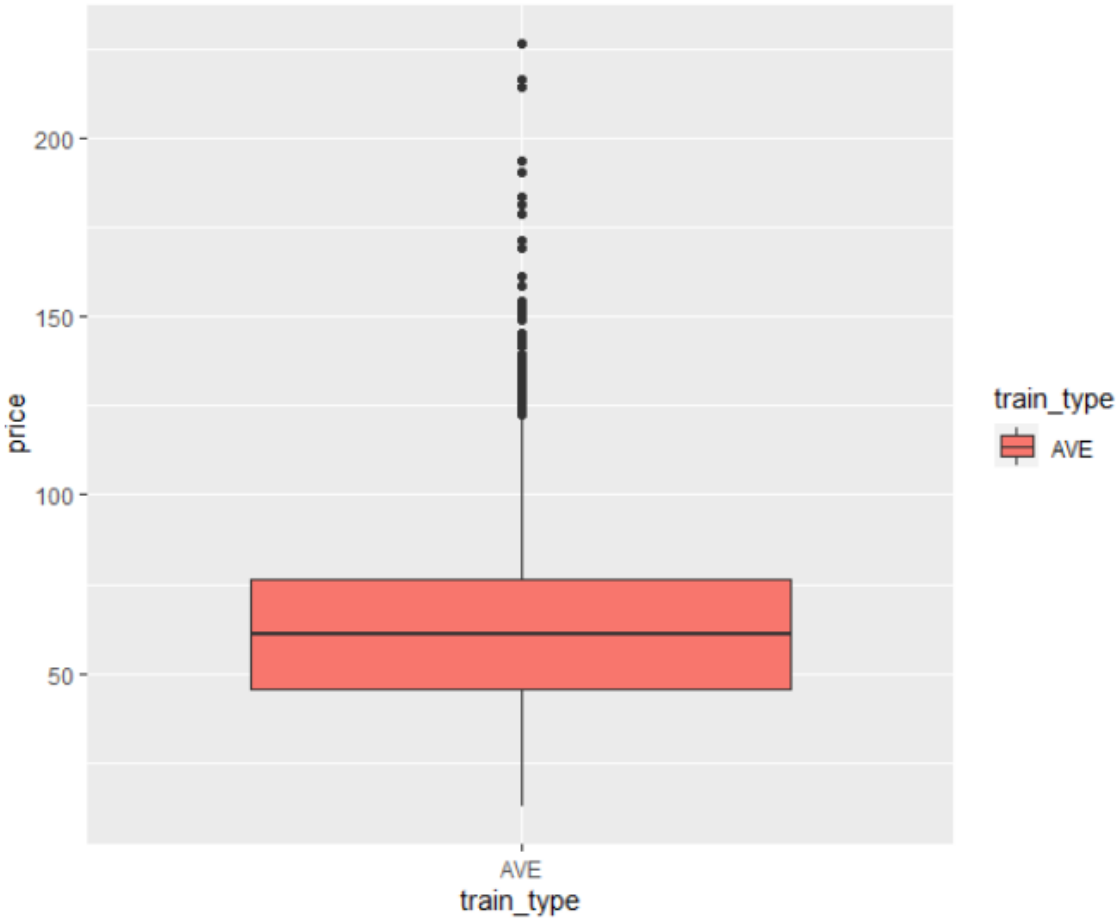



```
# 6.
# 마드리발 도착지별 가격 박스플롯으로
# 티켓가격의 높은 순을 확인해보자
str(madrid_origin)

ggplot(madrid_origin, aes(x=destination,y=price,fill=destination))+
  geom_boxplot()+
  coord_flip()


```

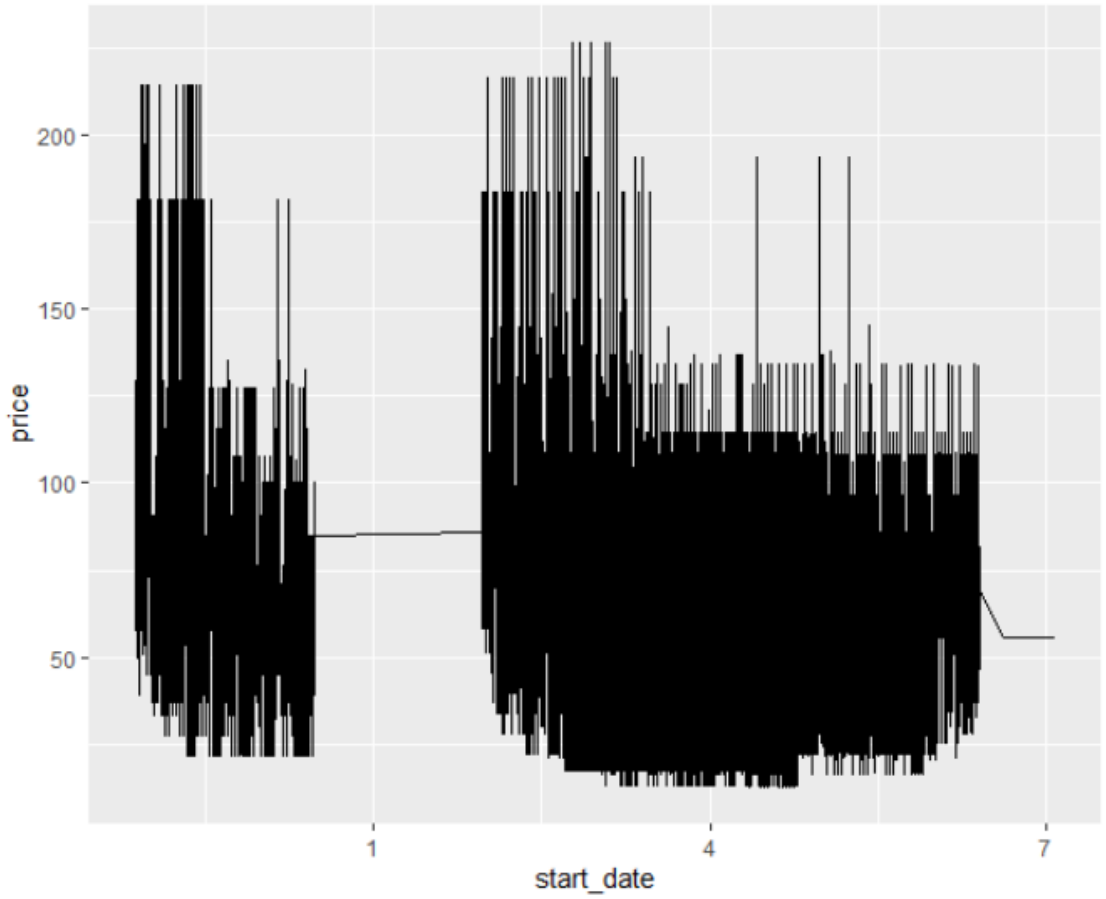



```r
# 7.
# AVE의좌석 등급별 가격박스플롯이 시각화
# 똑같은 열차와 똑같은 좌석등급, 똑같은 도착지라 하더라도 가격이 차이가 나는 것을 확인할 수 있다.

#전좌석 타입
ggplot(madrid_origin, aes(x=train_type ,y=price,fill=train_type))+
  geom_boxplot()+
  coord_flip()

#AVE로 시작되는 종류의 좌석 등급별 타입
ggplot(train_types, aes(x=train_type ,y=price,fill=train_type))+
  geom_boxplot()+
  coord_flip()

#AVE의 좌석 등급별 타입
ggplot(train_type_one, aes(x=train_type ,y=price,fill=train_type))+
  geom_boxplot()



subset(madrid_origin, train_type == c('AVE','AVE-AVANT','AVE-AVE','AVE-LD','AVE-MD','AVE-TGV'))

train_types<-subset(madrid_origin, train_type == 'AVE'|train_type == 'AVE-AVANT' |train_type == 'AVE-AVE'
       | train_type =='AVE-LD' | train_type =='AVE-MD' |train_type =='AVE-TGV' )

train_type_one<-subset(madrid_origin, train_type == 'AVE')
```


AVE-TGV
AVE-MD
AVE-LD
AVE-AVE
AVE-AVANT
AVE
train_type
50
100
150
200
price
train_type
AVE
AVE-AVANT
AVE-AVE
AVE-LD
AVE-MD
AVE-TGV

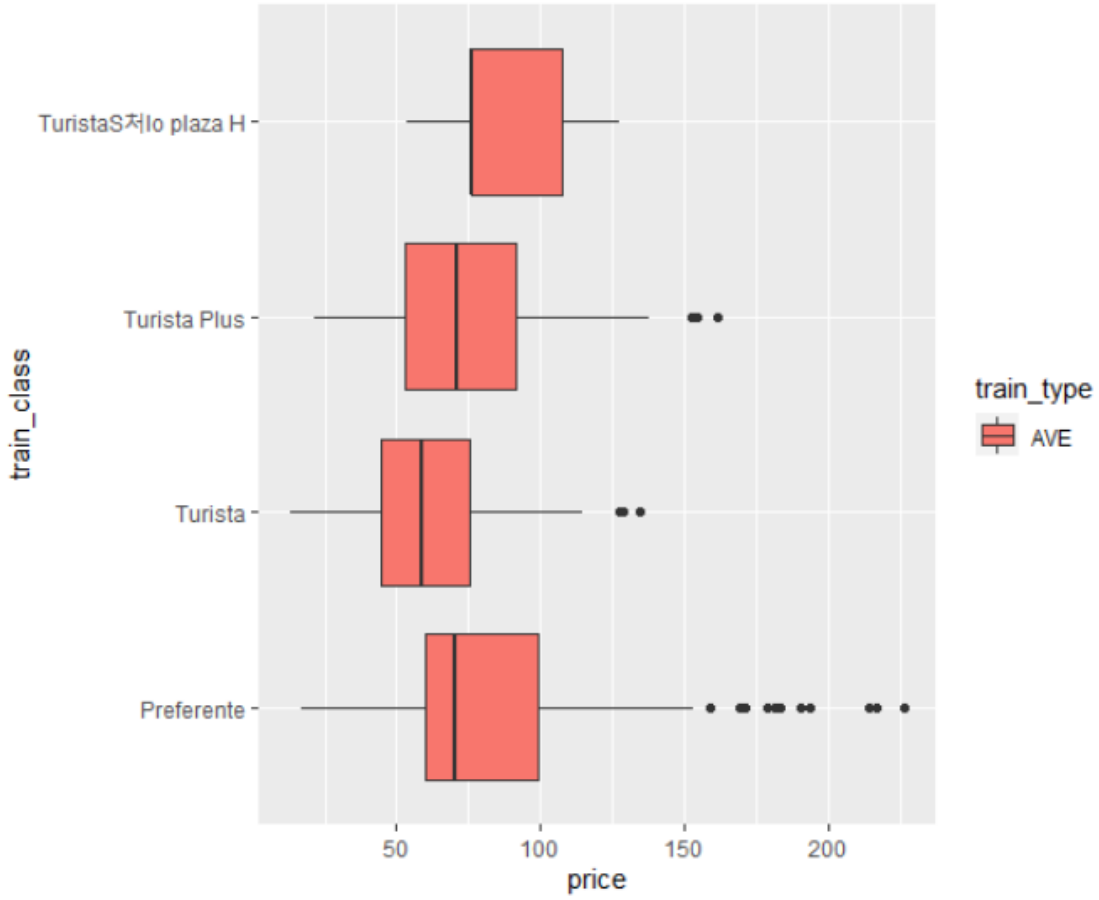

200
150
100
50
price
AVE
train_type
train_type
AVE

```r
# 8.
# 이 차이를 이해하고 싶어 시계열로 데이터를 만들어보았다.
str(train_type_one)

train_type_one$start_date <-as.POSIXct(train_type_one$start_date, format = "%Y-%m-%d %H:%M:%S")

ggplot(train_type_one, aes(x=start_date ,y=price,fill=start_date))+
  geom_line()

# 9.
#날짜 데이터 변환. as.POSIXct는 factor형식의 날짜 사용가능
```

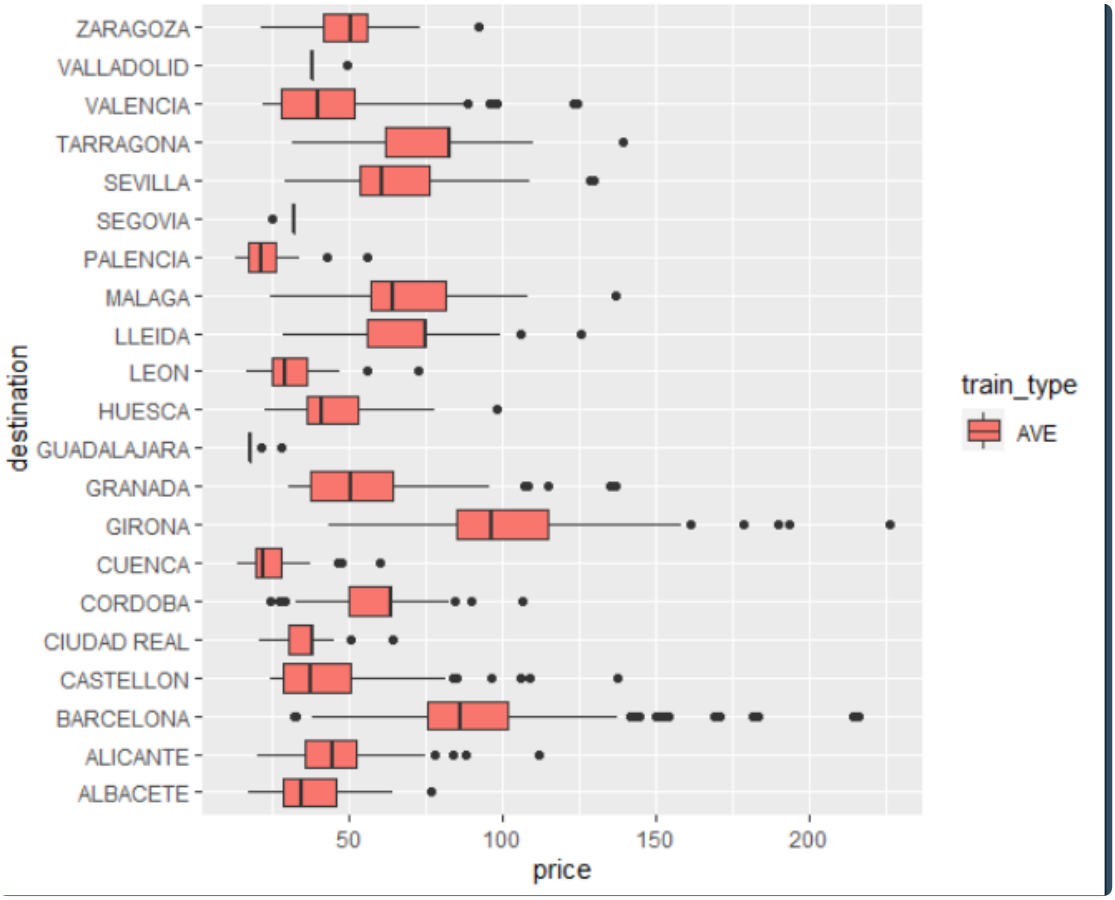



```r
# 11.
# 도착지별, 트레인 클래스별로 가격을 박스플롯형태로 나타낼 수도 있다.
ggplot(train_type_one, aes(x=train_type ,y=price,fill=train_type))+
  geom_boxplot()

ggplot(train_type_one, aes(x=destination ,y=price,fill=train_type))+
  geom_boxplot()
```

트레인 클래스별



목적지별 가격



'R' 카테고리의 다른 글□

[R] R에서 Database 사용하기 / DB 기본적인 구문 사용하기□

[R] 예제를 통한 데이터 전처리 작업□

[R] R을 통해 시계열 그래프 만들기 (자료 분석을 위한 시각화와 실습 예제)□

[R] 같은 형태의 ggplot 과 barplot 만들기 (차이 비교)□

[R] ggplot2 패키지 설치 에러시 해결 방법□

[R] R 을 활용한 데이터 탐색(Exploratory Data Analysis)□

R 시계열 그래프 R 시계열 그래프 만들기 시계열 그래프 시계열 그래프 그리기

꾸까꾸

혼자 끄적끄적하는 블로그 입니다.