

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

노트북: [TIL-MY]

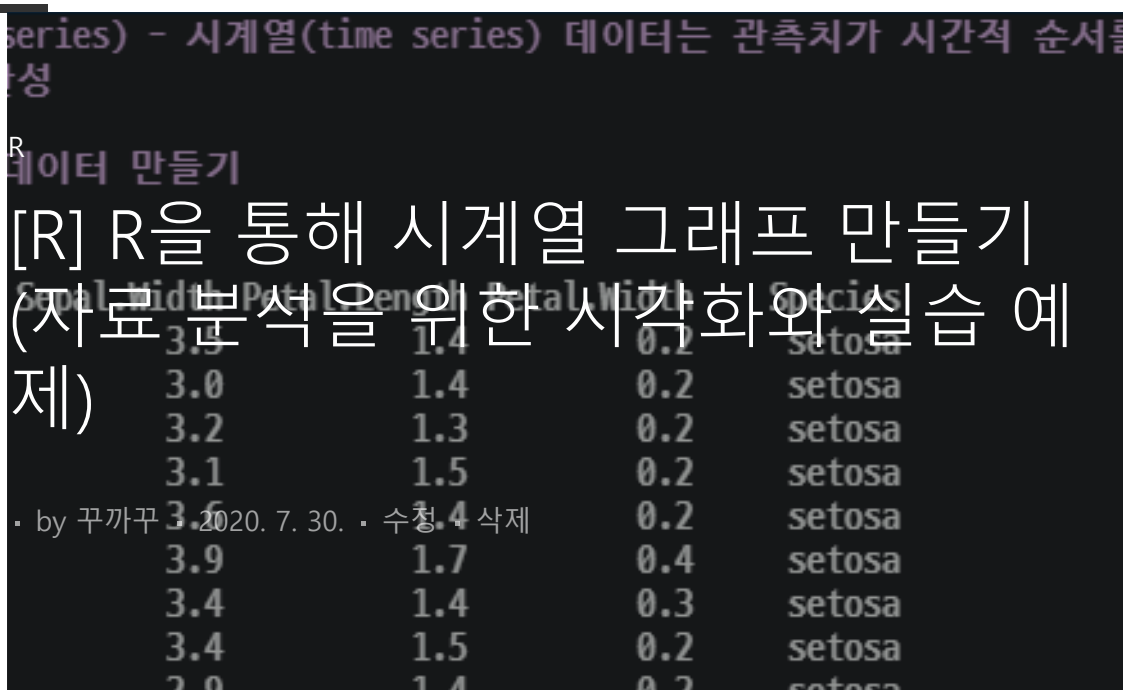
만든 날짜: 2020-08-02 오후 8:19

URL: <https://continuous-development.tistory.com/47>

나무늘보의 개발 블로그

홈

태그



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

분류 전체보기

Python

Database

ASP.NET

Algorithm

Deep learning

여기서는 시간의 흐름에 따라 값이 변화하는 것을 그 래프로 만드는 것을 목적으로 한다.

일단 첫 번째 iris는 시계열 데이터는 아닌 것 같다. 행 으로 나눠서 행이 변화하는 것에 따라서 그래프를 그 린다.

```
> # 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. 처음에 는 char데이터여서 이것을 타입 변환을 해줘 야 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
>
> rbind
>
> 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
6    6          5.4         3.9          1.7          0.4  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 irisD

AWS

ETC..

R 

공지사항

글 보실 때 주의사
항

: 최근글 : 인
기글

[Algorithm] 파이선
을 파..

2020.07.31

[R] R

을 ...



2020.07.30

[R]

같...



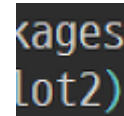
2020.07.30

[Algorithm] 프로그
래머스..

2020.07.30

[R]

gg...



2020.07.30

최근댓글

태그

프로그래머스 레

벨1 python,

프로그래머스 레

벨1,

F의 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 -Species
>
> colsColor <- topo.colors(4, alpha = .4) # rgb코드값을 랜덤하게 뽑아온다. 그다음 alpha는 투명도를 나타낸다.
> colsColor
[1] "#4CD0FF66" "#80E5FF66" "#80FF4D66" "#FFFF8066"
> 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, Petal.length, Petal.Width를 세로축으로 바꿔주는 작업을 한다. 그때 기준을 seq(행 번호)와 Species를 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 걸게 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.

인스턴스,
프로그래머스 스
킬체크,
행렬, 파이썬,
R 시계열 그래프
만들기,
R 시계열 그래프,
SQL, Python,
시계열 그래프,
rbind, AWS,
barplot 함수,
파이썬을 파이썬
답게,
cbind,
테이블 생성,
프로그래머스 스
킬체크 레벨1,
ggplot2 에러,
사용법,
시계열 그래프 그
리기,
날짜함수, 설정,
substr,
ggplot 함수,
DDL, ggplot,
Oracle SQL,
ggplot2 패키지
설치 에러,
Oracle

전체 방문자

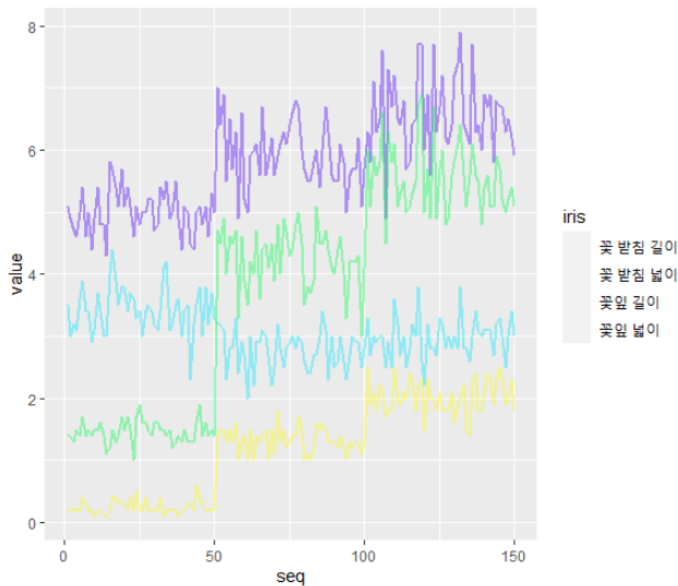
125

Today : 0

Yesterday : 0

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. 이 경우에는 날짜 데이터 타입으로 바꿔줘야 시계열 작업이 가능하므로 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
360

```

```

> 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

```

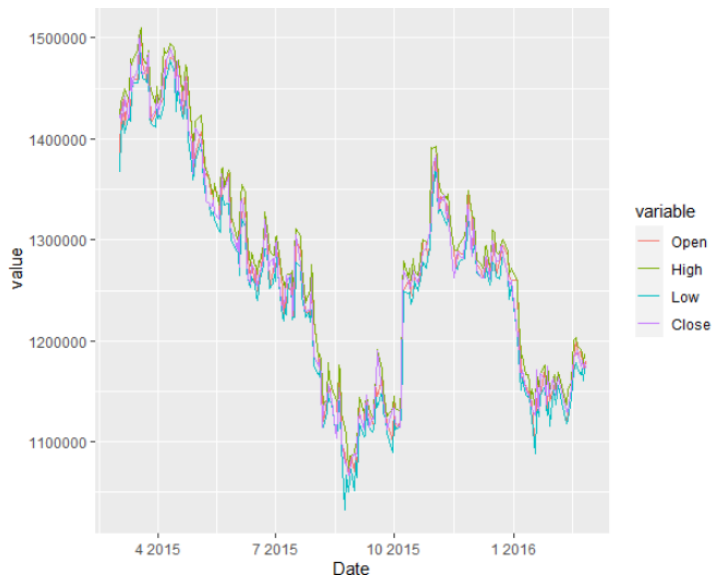
여기서 melt를 통해서 data와 볼륨에 따른 variable 값을 가지게 만든다. 우리가 찾을 것은 시간의 변화에 따른 variable(open,high,low,close)를 구하는 것이다.

```
> # 여기서 variable을 만들어야된다. melt로 data와 volume을 기준으로 값과 나머지 4가지 값인 open, high, low, close가 필요하다.
> # 그래서 melt로 id와 volume을 제외한 나머지 가로값을 세로값으로 만들기 위해 melt를
> # 사용하고 나오는 결과값을 보면 date에 따른 open, high, low, close이 생긴다.(Volume을 같이 묶은 이유는
> # 여기서의 나머지 4개의 값은 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	1283000
8	2016-02-17	245929	Open	1179000

```
367
368 #이걸 여기에 넣어주는데 x에 날짜를 넣고 y축에는 value를 넣어준다. 모든값에 대한 value이다 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 시계열이나 시간의 흐름에 따라 변화를 보기 위해
369 #날짜를 넣고 y축에는 variable에 대한 값을 보기 위해서 value를 넣어준다.
370 #그다음 col을 variable을 사용함으로써 group으로 묶어주는 효과와 동시에 같은 값끼리 같은 색을 나타내게
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

```
396 # 1.  
397 # 데이터 내에 결측치 여부를 확인한다.  
398 # NA값이 310681개 있는 것을 확인할 수 있다.  
399  
400 sum(is.na(trains))  
401 str(trains)  
402
```

```
403 # 2.  
404 # filter와 !is.na함수를 통해 결측치를 모두 제거했다.  
405 |  
406 trainsNa<-na.omit(trains)  
407 sum(is.na(trainsNa))  
408
```

```
409 # 3.  
410 # 마드리드 출발  
411 # 마드리드에서 출발하는 열차 데이터만을 떼어내 madrid_origin이라는 변수로 저장하고  
412 # 우선, 마드리드에서 출발하는 열차 데이터만을 이용해 비교해보기로 한다.  
413  
414 head(trainsNa)  
415 str(trainsNa)  
416 madrid_origin <- subset(trainsNa, origin=='MADRID')  
417 madrid_origin$origin  
418
```

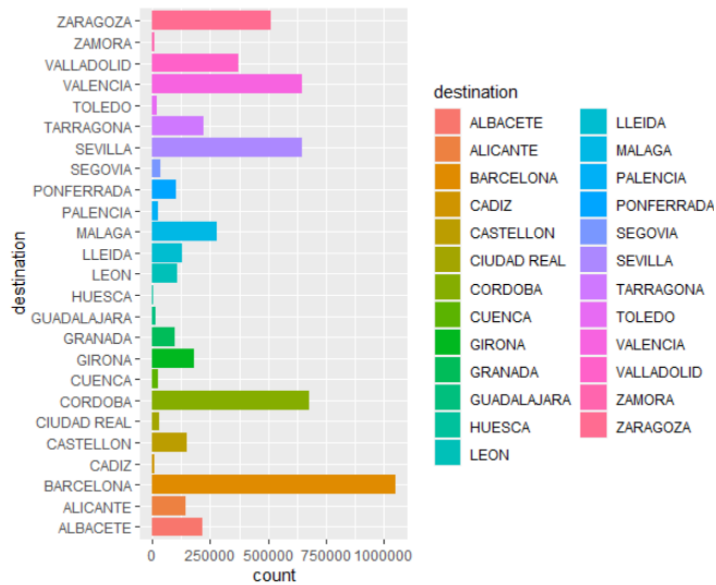
```
419 # 4.  
420 # summary함수를 통해 일반적 데이터 정보를 다시 확인한다.  
421 summary(madrid_origin)  
422
```

```
> 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.: 37.00	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. : 1226.40	

fare	price tree	batch	id
Length:5699800	Length:5699800	Length:5699800	Min. : 1
Class :character	Class :character	Class :character	1st Qu.: 2934004
Mode :character	Mode :character	Mode :character	Median : 5976612
			Mean : 6069570
			3rd Qu.: 8977866
			Max. : 19422154

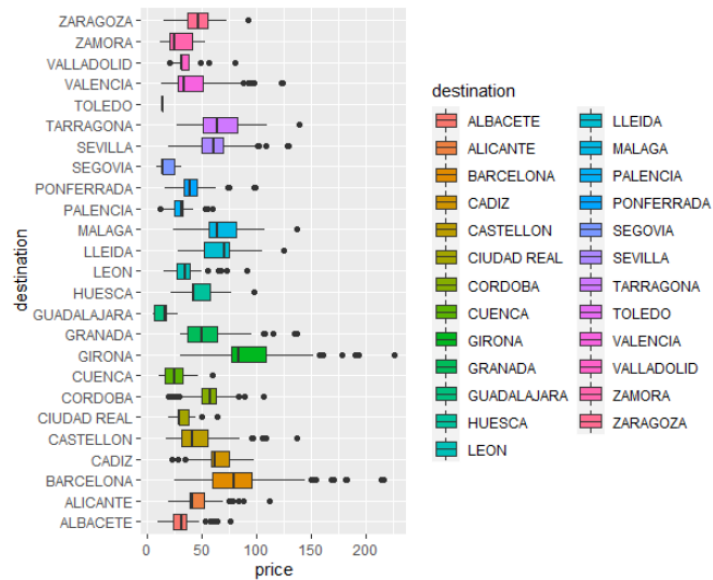
```
423 # 5.  
424 # 마드리드 출발 열차의 빈도 수  
425 # 마드리드를 출발하는 기차의 도착 도시별 운행빈도 수를 바탕으로 나타내보자  
426 ggplot(madrid_origin,aes(x=destination,fill=destination))+  
427   geom_bar()+  
428   coord_flip()
```



```

431 # 6.
432 # 마드리탈 도착지별 가격 박스플롯으로
433 # 티켓가격의 높은 순을 확인해보자
434 str(madrid_origin)
435
436 ggplot(madrid_origin, aes(x=destination,y=price,fill=destination))+
437   geom_boxplot()+
438   coord_flip()
439
440
441

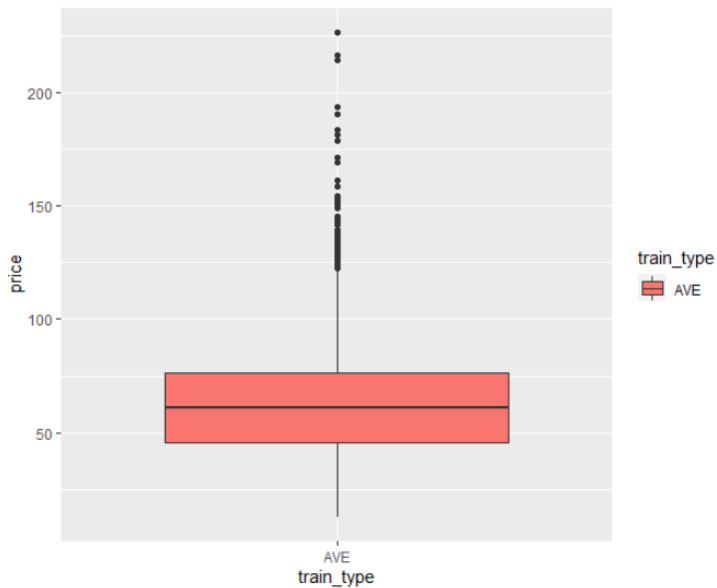
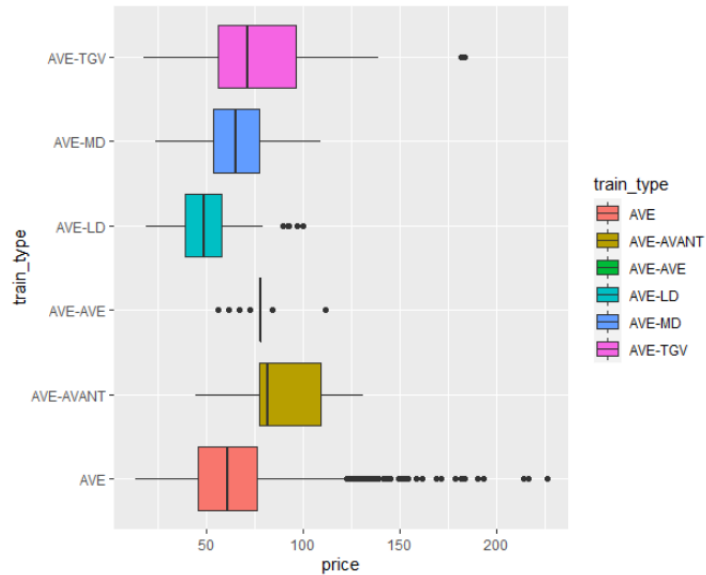
```




```

441 # 7.
442 # AVE의 좌석 등급별 가격박스플롯을 시각화
443 # 똑같은 열차와 똑같은 좌석등급, 똑같은 도착지라 하더라도 가격이 차이가 나는 것을 확인할 수 있다.
444
445 #전좌석 타입
446 ggplot(madrid_origin, aes(x=train_type, y=price, fill=train_type))+
447   geom_boxplot()+
448   coord_flip()
449
450 #AVE로 시작되는 종류의 좌석 등급별 타입
451 ggplot(train_types, aes(x=train_type, y=price, fill=train_type))+
452   geom_boxplot()+
453   coord_flip()
454
455 #AVE의 좌석 등급별 타입
456 ggplot(train_type_one, aes(x=train_type, y=price, fill=train_type))+
457   geom_boxplot()
458
459
460
461 subset(madrid_origin, train_type == c('AVE', 'AVE-AVANT', 'AVE-AVE', 'AVE-LD', 'AVE-MD', 'AVE-TGV'))
462
463 train_types<-subset(madrid_origin, train_type == 'AVE';train_type == 'AVE-AVANT';train_type == 'AVE-AVE'
464 ; train_type == 'AVE-LD' ; train_type == 'AVE-MD' ;train_type == 'AVE-TGV' )
465
466 train_type_one<-subset(madrid_origin, train_type == 'AVE')
467

```

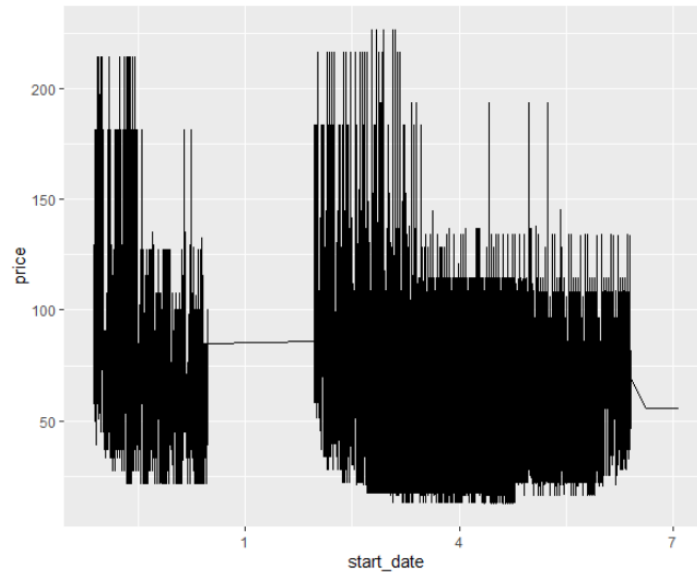


```
# 8.
# 이 작업을 이해하고 싶어 시계열로 데이터를 만들어보았다.
str(train_type_one)
```

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

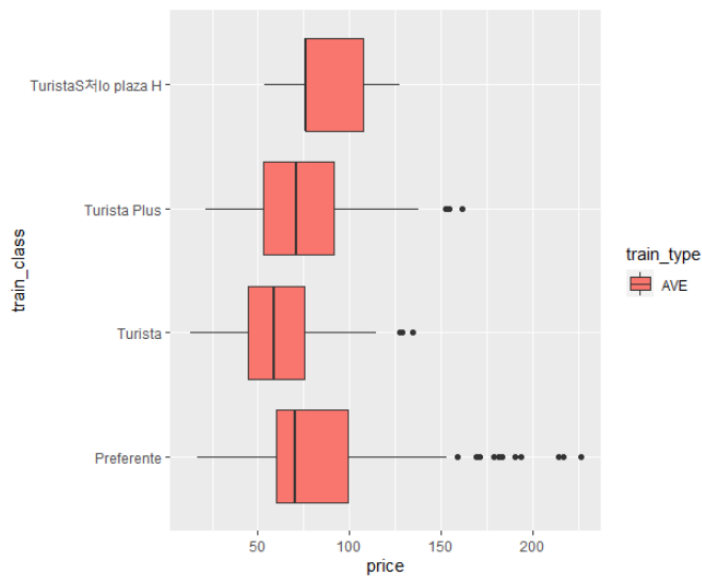
```
ggplot(train_type_one, aes(x=start_date, y=price, fill=start_date))+
  geom_line()
```

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

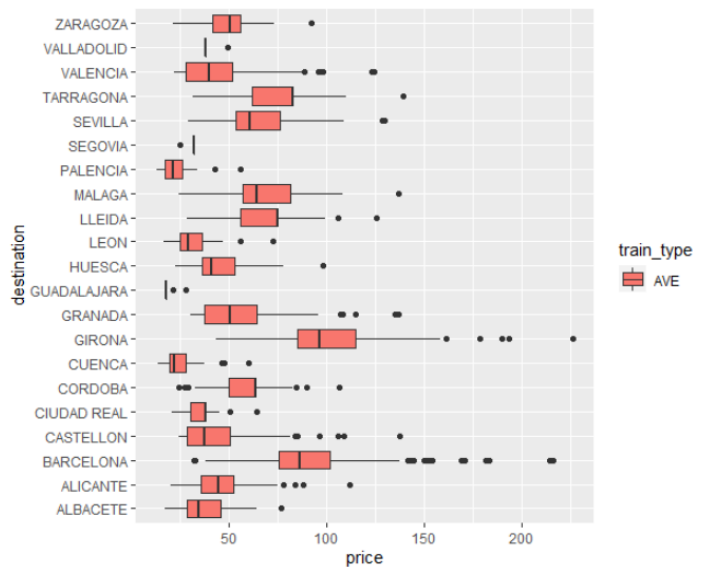


```
485
486 # 11.
487 # 도착지별, 트레인 클래스별로 가격을 박스플롯형태로 나타낼 수도 있다.
488 ggplot(train_type_one, aes(x=train_type, y=price, fill=train_type))+
489   geom_boxplot()
490
491 ggplot(train_type_one, aes(x=destination, y=price, fill=train_type))+
492   geom_boxplot()
```

트레인 클래스별



목적지별 가격



'R' 카테고리의 다른 글

[R] R을 통해 시계열 그래프 만들기
(자료 분석을 위한 시각화와 실습 예제) (0)

2020.07.30

[R] 같은 형태의 ggplot 과 barplot 만들기 (차이 비교) (0)	2020.07.30
[R] ggplot2 패키지 설치 에러시 해결 방법 (0)	2020.07.30
[R] R 을 활용한 데이터 탐색(Exploratory Data Analysis) (0)	2020.07.30
[R] R ggplot 사용법 (데이터 시각화 도구) (0)	2020.07.28
[R] R 에서 사용되는 기본적인 시각화 그래프-2 (0)	2020.07.28

태그

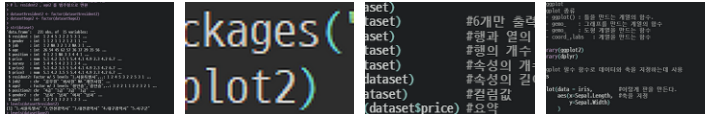
R 시계열 그래프

R 시계열 그래프 만들기

시계열 그래프

시계열 그래프 그리기

관련글



[R] 같은 형... [R] ggplot... [R] R 을 활... [R] R ggpl...

댓글 0



