

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

노트북: [TIL-MY]

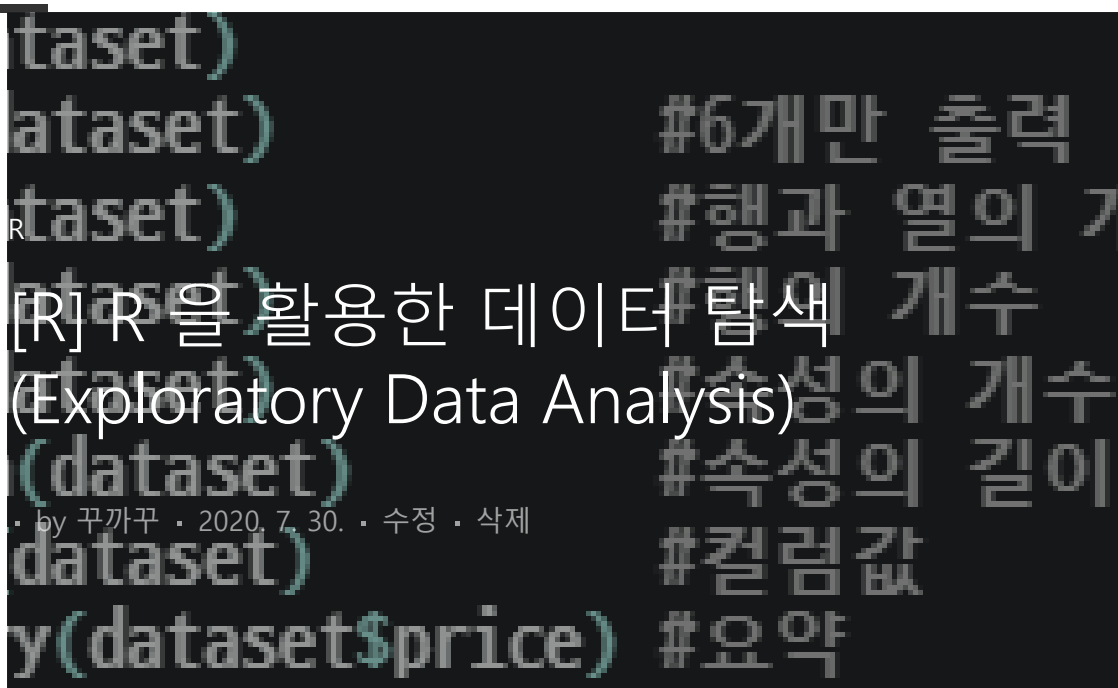
만든 날짜: 2020-07-31 오전 8:23

URL: <https://continuous-development.tistory.com/43?category=793392>

나무늘보의 개발 블로그

홈

태그



EDA(Exploratory Data Analysis)

수집한 데이터가 들어왔을 때, 이를 다양한 각도에서 관찰하고 이해하는 과정입니다. 한마디로 데이터

분류 전체보기 

Python

Database

ASP.NET

Algorithm 

Deep learning

를 분석하기 전에 그래프나 통계적인 방법으로 자료를 직관적으로 바라보는 과정입니다.

여기서는 7가지 단계로 EDA로 하겠습니다.

1.데이터 탐색

2.결측치(NA) 처리

3.이상치(outlier) 발견 처리

4.리코딩(코딩 변경)

5.파생변수, 가변수

6.시각화

7.의사결정

※여기서 의사결정 부분은 데이터를 통해 결정하는 과정으로 생략하겠습니다.

1.데이터 탐색

AWS

ETC..

R 

공지사항

글 보실 때 주의사항

: 최근글 : 인기글

[R] R
을 ...



2020.07.30

[R]
같...

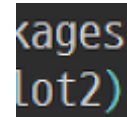


2020.07.30

[Algorithm] 프로그램
래머스..

2020.07.30

[R]
gg...



2020.07.30

[R] R
을 ...



2020.07.30

최근댓글

태그

테이블 생성,
시계열 그래프 그
리기,

데이터를 확인하는 작업

```
107 str(dataset)
108 head(dataset)      #6개만 출력
109 dim(dataset)        #행과 열의 개수
110 nrow(dataset)       #행의 개수
111 ncol(dataset)       #속성의 개수
112 length(dataset)     #속성의 길이
113 names(dataset)      #컬럼값
114 summary(dataset$price) #요약
```

위와 같은 명령어로 데이터를 확인한다.

2.결측치(NA) 처리

데이터 분석을 하는데 방해가 되는 NA 값을 처리한다.

결측치는 보통 중위수나 평균값 또는 0 / 삭제 이런 방식을 사용한다.

```
> #결측치 처리하는 방법
> # 중위수나 평균값으로 / 0으로 / 삭제
> # caret: na.omit() - 전체의 데이터셋을 대상으로 결측치를 처리한다.
> dataset_new <- na.omit(dataset)
> table(is.na(dataset_new))

FALSE
1463
```

```
118
119 #결측치 처리하는 방법
120 # 중위수나 평균값으로 / 0으로 / 삭제
121
122 # caret: na.omit() - 전체의 데이터셋을 대상으로 결측치를 처리한다.
123 dataset_new <- na.omit(dataset)
124 table(is.na(dataset_new))
125 str(dataset_new)
126
127 # 평균
128 mean(dataset$price, na.rm = T)
129 |
130 # 0으로 대체
131 price <- ifelse(is.na(dataset$price), 0, dataset$price)
132
133 # 평균으로 대체
134 price <- ifelse(is.na(dataset$price), mean(dataset$price, na.rm = T), dataset$price)
135
136 # 통계적 방법
137
```

substr,
R 시계열 그래프,
ggplot2 에러,
설정,
barplot 함수,
Oracle,
프로그래머스 레벨1 python,
날짜함수,
ggplot 함수,
프로그래머스 레벨1,
Oracle SQL,
rbind, SQL,
ggplot,
시계열 그래프,
인스턴스,
프로그래머스 스킬체크 레벨1,
프로그래머스 스킬체크,
ggplot2 설치 에러,
ggplot2 설치 오류,
ggplot2 패키지 설치 에러,
사용법,
ggplot2 설치 안됨,
DDL, 행렬, AWS,
cbind,
R 시계열 그래프 만들기

전체 방문자

```

143 # type : 1, 2, 3
144 # TYPE이 1일 때는 price 값을 15%/10%/5% 이렇게 줄인다.
145
146 # 300 고객유형을 판단하여 새로운 가격수 pricesta 만들어 처리된 값을 넣어보자
147 # type에 따라 평균값을 넣어주는 구간 / 평균값을 구하는데 있어서 결측치는 제외하고 넣는다.
148
149 for (i in 1:nrow(dataset_new)){
150   if(dataset_new$type[i] == 1){
151     dataset_new$pricesta[i] <- mean(dataset$price, na.rm = T) * 1.15
152   }else if(dataset_new$type[i] == 2){
153     dataset_new$pricesta[i] <- mean(dataset$price, na.rm = T) * 1.10
154   }else{
155     dataset_new$pricesta[i] <- mean(dataset$price, na.rm = T) * 1.05
156   }
157 }
158
159 #type 에 따라 price 의 값을 넣어주는 구간 / 여기서 결측치들을 해서 값이 없으면 평균을 넣고 있을 경우에는 price를 넣는다.
160 for (i in 1:nrow(dataset_new)){
161   if(dataset_new$type[i] == 1){
162     dataset_new$pricesta[i] <- (ifelse(is.na(dataset_new$price[i]), mean(dataset_new$price[i], na.rm = T), dataset_new$price[i])) * 1.15
163   }else if(dataset_new$type[i] == 2){
164     dataset_new$pricesta[i] <- (ifelse(is.na(dataset_new$price[i]), mean(dataset_new$price[i], na.rm = T), dataset_new$price[i])) * 1.10
165   }else{
166     dataset_new$pricesta[i] <- (ifelse(is.na(dataset_new$price[i]), mean(dataset_new$price[i], na.rm = T), dataset_new$price[i])) * 1.05
167   }
168 }
169
170 dataset_new

```

```

219 #예제
220 # age 체크
221 summary(dataset$age)
222
223 # age 결측치 제거 후 boxplot으로 시각화
224
225 subset(dataset$age, na.rm=T)
226
227 # 결측치 날리는 방법
228 dataset_new2 <- na.omit(dataset$age) |
229 dataset_new3 <- dataset$age[!is.na(dataset$age)]
230
231 summary(dataset_new2)
232
233 boxplot(dataset_new2, horizontal= T)
234

```

3.이상치(outlier) 발견 처리

결측치를 처리한 후에는 이상치를 발견 후 처리해야 된다. 이 데이터들은 분석을 하는데 있어서 악영향을 미친다.

1)변수 유형이 이산변수인경우

```

# 3.이상치(outlier) 발견 처리

gender <- dataset$gender
range(gender)
table(gender)

```

gender라는 성별의 속성을 확인하고 있다.

```

> gender
[1] 1 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 2 2 2 2 1 2 2 2 1 2 1 2
(77) 2 2 2 1 1 2 2 2 1 1 2 1 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 2
(151) 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 2 2 2 1
(225) 2 2 2 1 2 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2 1 2 1 2 2 2 2 0 1 5 1 1 0 1
> range(gender)
[1] 0 5
> table(gender)
gender
0    1    2    5
2 173 124    1

```

gender는 범주형 데이터로서 형태를 봤을때 1,2 가 아닌 이상치인 0과 5가 들어있다 이걸 지워줘야한다.

```

189 # subset을 활용하여 이상치를 제거한 후 gender를 범주형으로 변환해보자
190 dataset <- subset(dataset, gender != 1 & gender != 2) # subset을 통해 gender가 1 or 2인 값만 가져와서 다시 dataset에 넣어준다.
191 dataset$gender <- factor(dataset$gender) # factor 함수를 통해 데이터를 factor로 바꾼다.
192 levels(dataset$gender) # levels 함수를 통해 해당 컬럼의 레벨이 어떻게 되어있는지 확인한다.
193 table(dataset$gender) # table 함수로 값이 어떻게 들어있는지 확인한다.
194 mtc<-table(dataset$gender)) # mtc로 저장 후 확인한다.
195 dim(dataset) # 행과 열의 개수를 확인한다
196 str(dataset)

```

위에서는 1과2인 값을 추출해서 다시 넣어주는 방식을 통해 이상치를 제거했다.

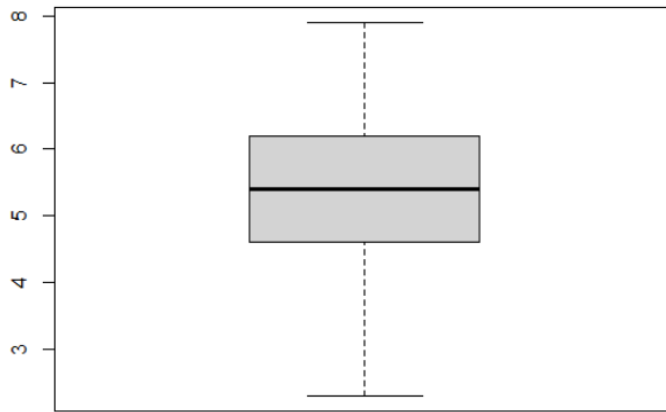
2) 변수 유형이 연속 변수 인 경우

```

> # 변수의 유형이 연속변수인 이상치 제거
>
> seqPrice <- dataset$price
> length(seqPrice)
[1] 297
> summary(seqPrice)
   Min.   1st Qu.   Median     Mean   3rd Qu.     Max.    NA's
-457.200   4.400   5.400   8.784   6.300  675.000     30
> #IQR (Q3-Q1) 6.2 - 4.6 = 1.6
> boxplot(seqPrice) # 시각화 해서 이상치 값을 확인
>

```

위의 속성을 summary를 했을때 사분위를 확인 했을때 이상치가 들어있는 것을 볼 수 있다.



위와같이 이상치가 제거된 걸 볼 수있다.

4.데이터 리코딩 / 파생변수, 가변수

아래에 있는 resident라는 속성값은 연속형으로 나타나 있지만 가독성을 위해서 범주형 데이터로 바꿔주었다.

그리고 이 데이터를 새로운 파생변수 resident_new라는 곳에 넣어주었다.

```

236 # 리코딩 - 데이터의 가독성을 위해서
237 # 연속형 → 범주형
238 # 형식) dataset$컬럼[조건식] <- 추가할 값
239 # 1 : 서울 , 2: 부산, 3: 광주, 4: 대전, 5 : 대구
240 head(dataset$resident)
241 summary(dataset$resident)
242
243 dataset$resident[is.na(dataset$resident)]
244
245 dataset$resident_new[dataset$resident==1] <- "서울"
246 dataset$resident_new[dataset$resident==2] <- "부산"
247 dataset$resident_new[dataset$resident==3] <- "광주"
248 dataset$resident_new[dataset$resident==4] <- "대전"
249 dataset$resident_new[dataset$resident==5] <- "대구"
250 dataset$resident_new
251
252 #주거지의 NA 값을 행정수도인 대전으로 대체
253
254 dataset$resident_new[is.na(dataset$resident)] <- "대전"
255 dataset$resident_new
256
257 dataset$resident_new <- factor(dataset$resident_new)
258 levels(dataset$resident_new)
259 str(dataset$resident_new)
260

```

#--간단한 예제 1

```

278 url <- "https://www.dropbox.com/s/bfjcm62zdt6/example_salary.csv?dl=1"
279 salary_data_eda <- read.csv(url, stringsAsFactors=F, na="-") # > 이 없으면 na로 취급한다.
280
281 # 1.컬럼명을 영문으로 변경
282 names(salary_data_eda) <- c("age", "Monthly salary", "Annual special salary amount", "Working hours", "The number of workers", "Career classification", "gender")
283 names(salary_data_eda)
284
285 # 2.각 컬럼별 결측값 확인
286 is.na(salary_data_eda$age)
287 is.na(salary_data_eda$Monthly salary)
288 is.na(salary_data_eda$Annual special salary amount)
289 is.na(salary_data_eda$Working hours)
290 is.na(salary_data_eda$The number of workers)
291 is.na(salary_data_eda$Career classification)
292 is.na(salary_data_eda$gender)
293
294 salary_data_eda$age[is.na(salary_data_eda$age)]
295 salary_data_eda$Monthly salary[is.na(salary_data_eda$Monthly salary)]
296 salary_data_eda$Annual special salary amount[is.na(salary_data_eda$Annual special salary amount)]
297 salary_data_eda$Working hours[is.na(salary_data_eda$Working hours)]
298 salary_data_eda$The number of workers[is.na(salary_data_eda$The number of workers)]
299 salary_data_eda$Career classification[is.na(salary_data_eda$Career classification)]
300 salary_data_eda$gender[is.na(salary_data_eda$gender)]
301
302 # 3.임금 평균 확인
303 salary_mean <- mean(salary_data_eda$Monthly salary, na.rm=T)
304 salary_mean
305
306
307 # 4.임금 중앙값 확인
308 summary(salary_data_eda$Monthly salary)
309 salary_median <- median(salary_data_eda$Monthly salary, na.rm=T)
310 salary_median
311
312 # 5.임금 범위 구해보기(최저,최고)
313
314 range(salary_data_eda$Monthly salary, na.rm=T)
315
316 #범위에 들어있는 값 구하기
317 salaryRange <- subset(salary_data_eda$Monthly salary,
318                       salary_data_eda$Monthly salary >= range(salary_data_eda$Monthly salary, na.rm=T)[1]
319                       & salary_data_eda$Monthly salary <= range(salary_data_eda$Monthly salary, na.rm=T)[2])
320
321 salaryRange
322
323 #최소 최대 범위를 구하는 range
324 range(salary_data_eda$Monthly salary, na.rm=T)[1]
325
326 # 6.임금에 대한 사분위수(quantile())
327 quantile(salaryRange)
328
329 # 7.성별에 따른 임금 격차 확인해보기
330 genderGapSalary <- salary_data_eda %>%
331   select(gender, Monthly salary) %>%
332   group_by(gender) %>%
333   summarise(salary = mean("Monthly salary", na.rm=T))
334
335 genderGapSalary

```



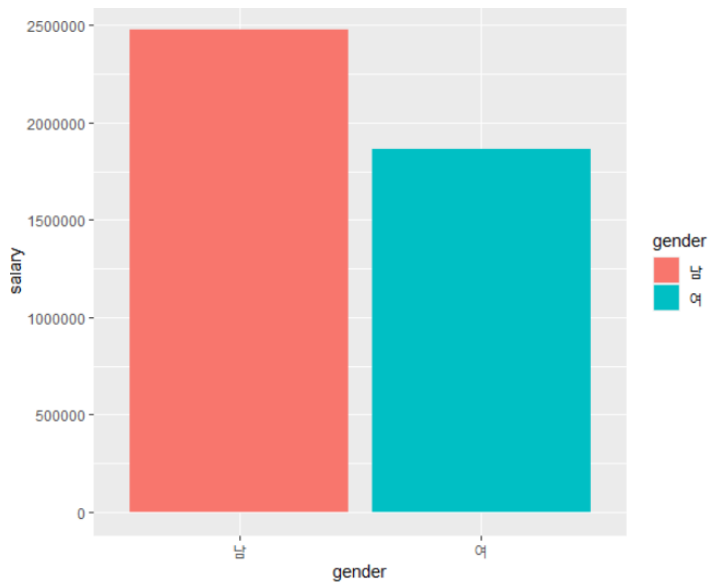
```

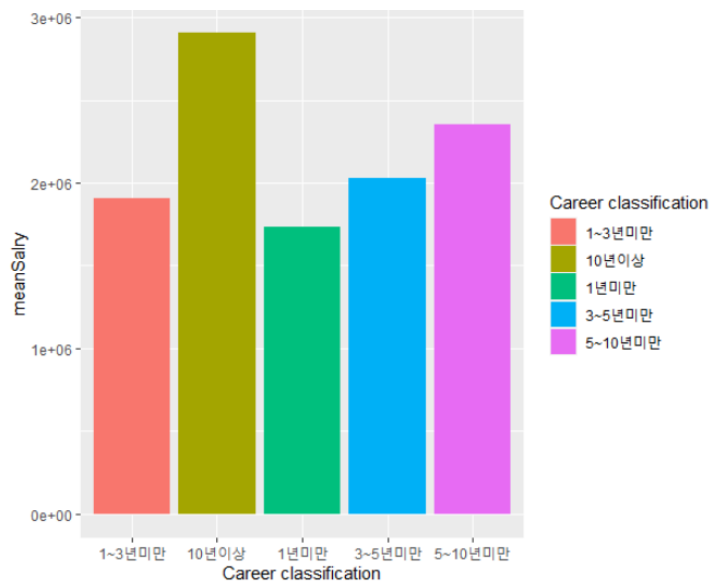
336 # 8. 분석한 데이터를 가지고 원하는 시각화 진행
337 ggplot(genderGap, aes(x=gender, y=salary, fill=gender)) +
338   geom_col()
339
340 # 9. 성별에 따른 표준편차 구하기
341 genderGapSd <- salary_data_eda %>%
342   select(gender, "Monthly salary") %>%
343   group_by(gender) %>%
344   summarise(sd = sd("Monthly salary", na.rm=T))
345 genderGapSd
346
347 # 10. 경력별 임금 평균치
348 CareerMeanSalary <- na.omit(salary_data_eda) %>%
349   select("Monthly salary", "Career classification") %>%
350   group_by("Career classification") %>%
351   summarise(meanSalary = mean("Monthly salary"))
352 CareerMeanSalary
353
354 # 11. 경력별 임금 평균치 시각화
355 ggplot(CareerMeanSalary, aes(x="Career classification", y=meanSalary, fill="Career classification")) +
356   geom_col()
357
358
359
360

```

6. 시각화

이렇게 구한 데이터를 원하는 형태의 차트나 도표로 시각화해서 보여준다.





#--간단한 예제 2

```

364 # 2차 실습
365 install.packages("readxl")
366 library(readxl)
367 ck <- read_xlsx(file.choose())
368 head(ck)
369
370
371 pattern <- regexpr("[가-힣]{1,4}동", ck$소재지전체주소)
372 dong <- regmatches(ck$소재지전체주소, pattern)
373 dong
374
375
376 # 실행결과 동 이름이 3글자인 경우와 4글자인 경우가 있으므로 지정한 자리만큼
377 # 글자를 추출하면 3글자인 동은 숫자가 포함된다.
378 # 공백과 숫자를 제거하자
379
380 a <- substr(ck$소재지전체주소, start = 12, stop = 16)
381
382
383 #첫번째 방법 - 군데이전 총칭로 2가에서 2라는 숫자를 지워서 안된다. p259 쪽
384 dong <- gsub("[0-9]${[[:blank:]]}", "", a)
385
386
387 #두번째 방법
388 dong <- sub(pattern = "[0-9]+${}+", # sub 로 $표현식(끝부분)으로 공백인 부분과 숫자인 부분을 뺀다
389 replacement = "",
390 x = a) # 정규표현식을 이용해서 모든숫자를 제거한다.
391

```

```

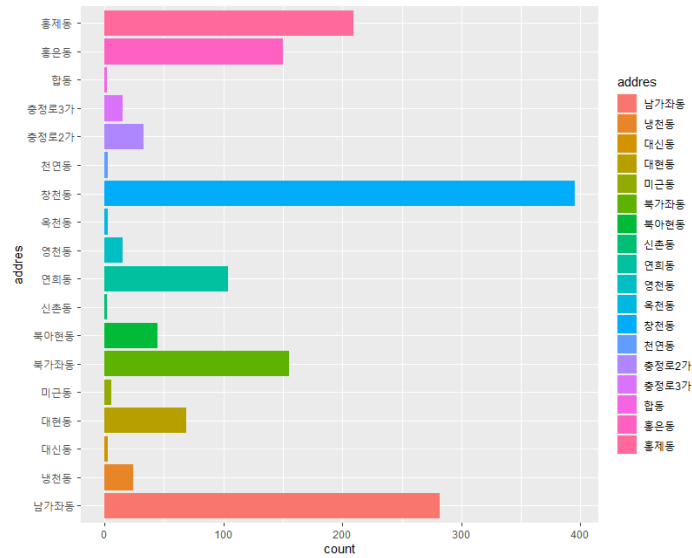
393 # 동별 도수분포표 만들어보기
394 # table() 함수를 이용해서 숫자 세기, 백수가 한개일때 도수분포표를 만들어줌
395 # 도수분포표값 합작별 개수를 나타낸 것이다
396
397 #첫번째 방법
398 dong<-data.frame("dong" = dong) #일단 데이터 프레임은 만들었다, 그래야 타입이 생기고
399 str(dong)
400 class(dong)
401
402 dong$dong <- factor(dong$dong) #장간 타입을 factor로 바꿔줬다.
403
404 dong
405 levels(dong$dong) # 그러면 레벨에도 분류가 되고
406 address<-data.frame(table(dong$dong)) # 여기서 테이블로 왔을때 카운트를 넣을 수 있기 때문이다, 이상태로 데이터 프레임은 다시 만들었다.
407 names(address) <- c('address','count') # 이름을 새로 지정해주고
408 address
409
410
411 #두번째 방법
412 address2<-d %>% group_by(dong) %>%
413   summarise(count=n())
414
415 address2
416 str(address2)
417
418 install.packages("dplyr")
419 library(dplyr)
420
421
422 ggplot(address, aes(x=address,y=count,fill=address ))+ # 여기서 ggplot으로 그림을 그린다.
423   geom_col() +
424   coord_flip()
425

```

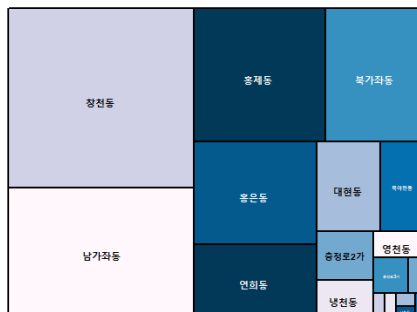
```

426
427 # 트리맵은 순서으로 데이터 프레임을 입력받는다.
428 # treemap(데이터, index=구분 할 , vColor=분포 할 , vColor=컬러, title=제목
429 library(treemapify)
430 library(ggplot2)
431
432
433 install.packages("treemap")
434 library(treemap)
435
436 # 여기서 index는 x축 처럼 값을 값을 vsize에 개수를 나타낸다. 개수에 따라서 사이즈가 달라진다. vColor는 fill과 같고 마지막 title로 마무리한다.
437 treemap(address, index=address , vsize='count' , vColor=address , title='동별 동합계 개수 , palette='huber')
438
439

```



동별 동합계 개수



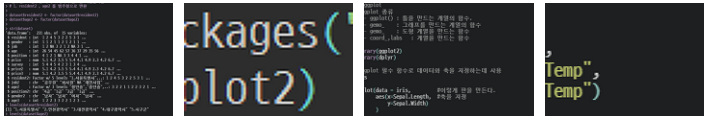
'R' 카테고리의 다른 글

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

태그

[eda](#)[Exploratory Data Analysis](#)[ploratory Data Analysis](#)[R을 통한 데이터 탐색](#)[R을 활용한 데이터 탐색](#)[데이터 탐색](#)

관련글



[R] 같은 형... [R] ggplot... [R] R ggpl... [R] R 에서 ...

댓글 0

TEL. 02.1234.5678 / 경기 성남시 분당구 판교역로

© Kakao Corp.

