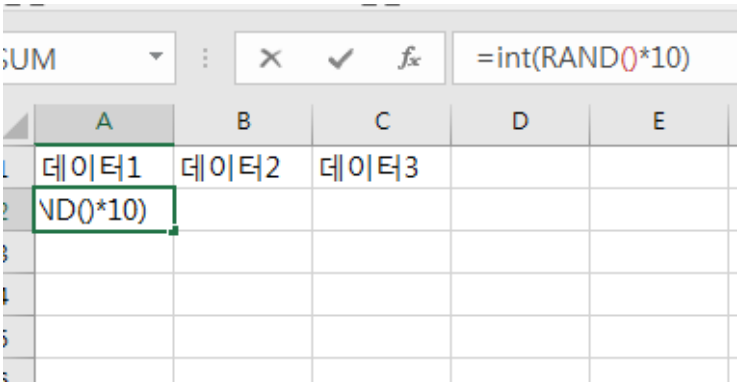
차트

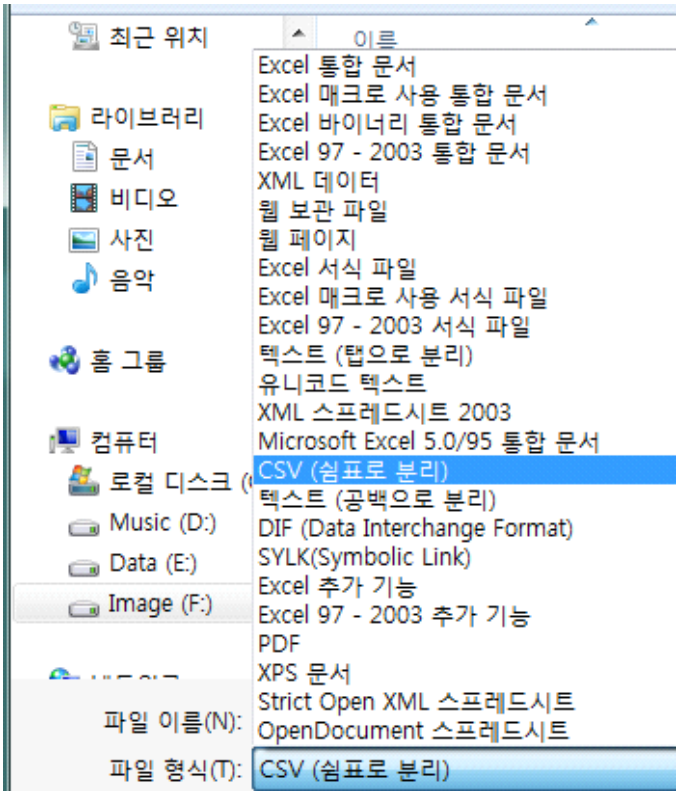
2017년 10월 21일 토요일 오전 10:10

데이터 만들기

2017년 10월 21일 토요일 오전 10:10



data1	data2	data3
3	1	4
6	0	7
0	1	1
6	1	9
8	7	5
5	4	8
7	7	8
8	0	9
6	2	6



막대그래프

2017년 10월 21일 토요일 오전 10:16

```
jin@jin-VirtualBox:~$ ssh -X jinuser@master
The authenticity of host 'master (192.168.56.1
ECDSA key fingerprint is SHA256:e7bXmod9+QBxq
```

```
jinuser@jin-VirtualBox:~$ sudo cp /media/sf_Share/Ex01.csv ./
[sudo] password for jinuser:
jinuser@jin-VirtualBox:~$ ls
Desktop             examples.desktop     protobuf-2.4.1
Ex01.csv            ga_mapper.R          protobuf-2.4.1.tar.gz
R                   ga_reducer.R         rmr2_3.3.1.tar.gz
```

```
jinuser@jin-VirtualBox:~$ ls -l Ex01.csv
-rwxr-x--- 1 root root 88 10월 21 10:19 Ex01.csv
jinuser@jin-VirtualBox:~$ sudo chown jinuser:hadoop Ex01.csv
[sudo] password for jinuser:
jinuser@jin-VirtualBox:~$ ls -l Ex01.csv
-rwxr-x--- 1 jinuser hadoop 88 10월 21 10:19 Ex01.csv
jinuser@jin-VirtualBox:~$
```

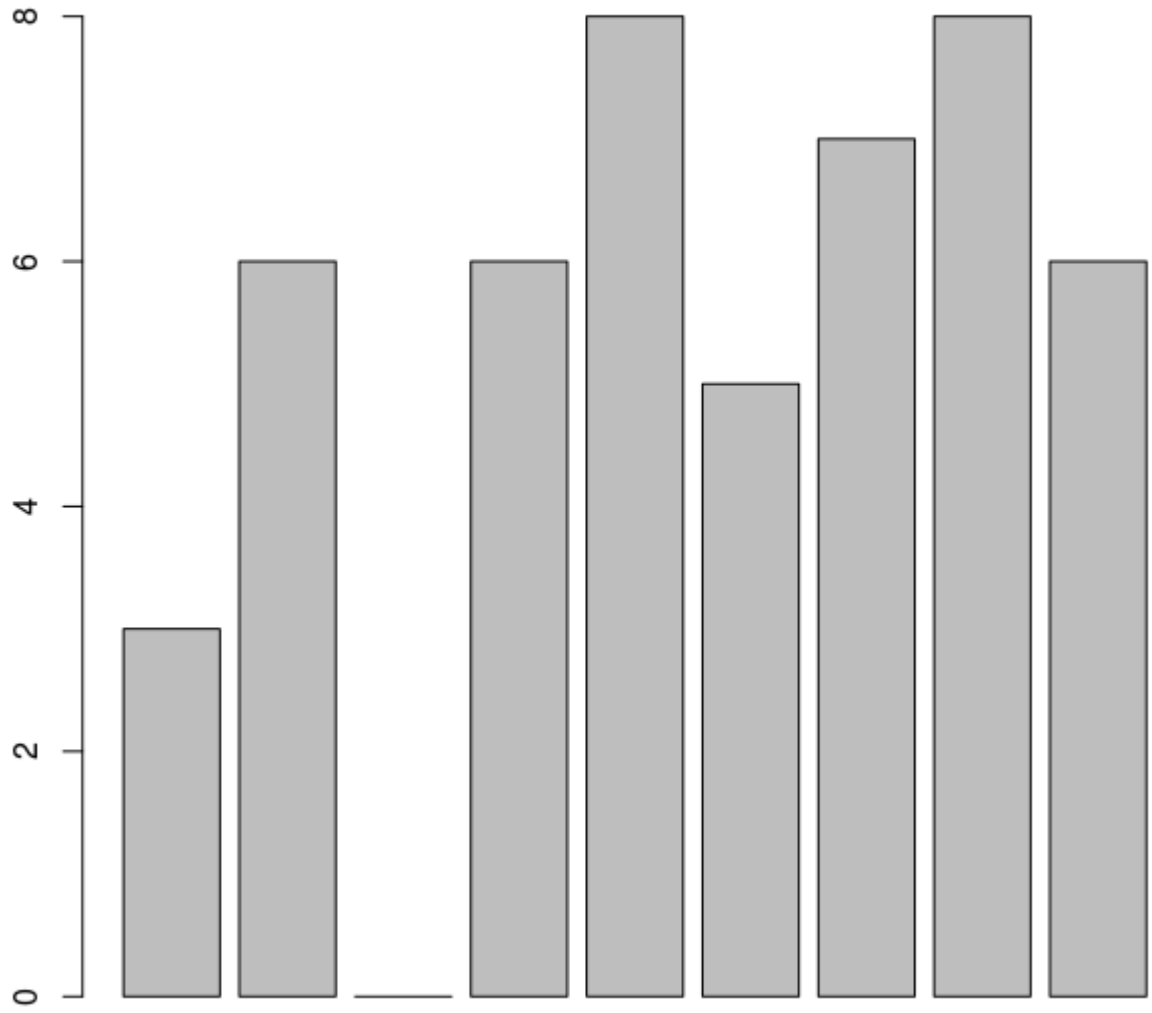
```
jinuser@jin-VirtualBox:~$ rstudio &
```

```
> dataVector<-read.csv("Ex01.csv", header=FALSE, sep=",")
> dataVector
      V1    V2    V3
1  data1 data2 data3
2      3     1     4
3      6     0     7
4      0     1     1
5      6     1     9
6      8     7     5
7      5     4     8
8      7     7     8
9      8     0     9
10     6     2     6
```

```
> dataVector<-read.csv("Ex01.csv", header=TRUE, sep=",")
> dataVector
  data1 data2 data3
1     3     1     4
2     6     0     7
3     0     1     1
4     6     1     9
5     8     7     5
6     5     4     8
7     7     7     8
8     8     0     9
9     6     2     6
```

Header가 존재할 경우 header를 제외하고 나머지를 데이터로 처리함.

```
barplot(dataVector$data1)
```



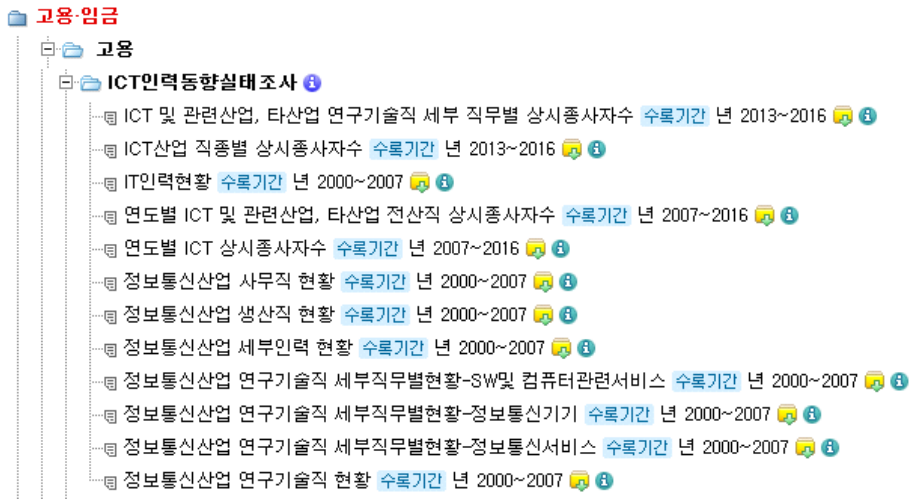
datavector에 존재하는 값 중 data1에 대한 막대 그래프 출력

점 그래프

2017년 10월 21일 토요일 오후 12:06

http://kosis.kr/index/index.jsp

고용, 임금 선택



정보통신산업 연구기술직 세부직무별 현황 선택

일괄설정 + | 항목[1/1] | IT산업직무별[10/10] | 시점[3/8] | 통계표조회

(단위 : 명) 새창보기 › | URL › |

IT산업직무별	2007	2006	2005
SI개발	15,541	13,216	13,254
SW개발	36,263	33,648	31,936
디지털콘텐츠	2,790	3,127	2,859
시스템운영관리	10,759	14,554	13,819
통신/방송서비스	797	1,302	1,197
HW개발	1,992	1,155	1,097
HW유지관련	7,571	8,776	7,813
IT관련교육	399	419	344
IT기술영업	3,586	4,054	3,834
합 계	79,698	80,251	76,153

파일 다운로드


```
jinuser@jin-VirtualBox:~$ sudo cp /media/sf_Share/정보통신산업_연구기술직_세부직무별현황SW및_컴퓨터관련서비스_20171021120848.csv ./
```

```
jinuser@jin-VirtualBox:~$ sudo mv 정보통신산업_연구기술직_세부직무별현황SW및_컴퓨터관련서비스_20171021120848.csv Ex01.csv
```

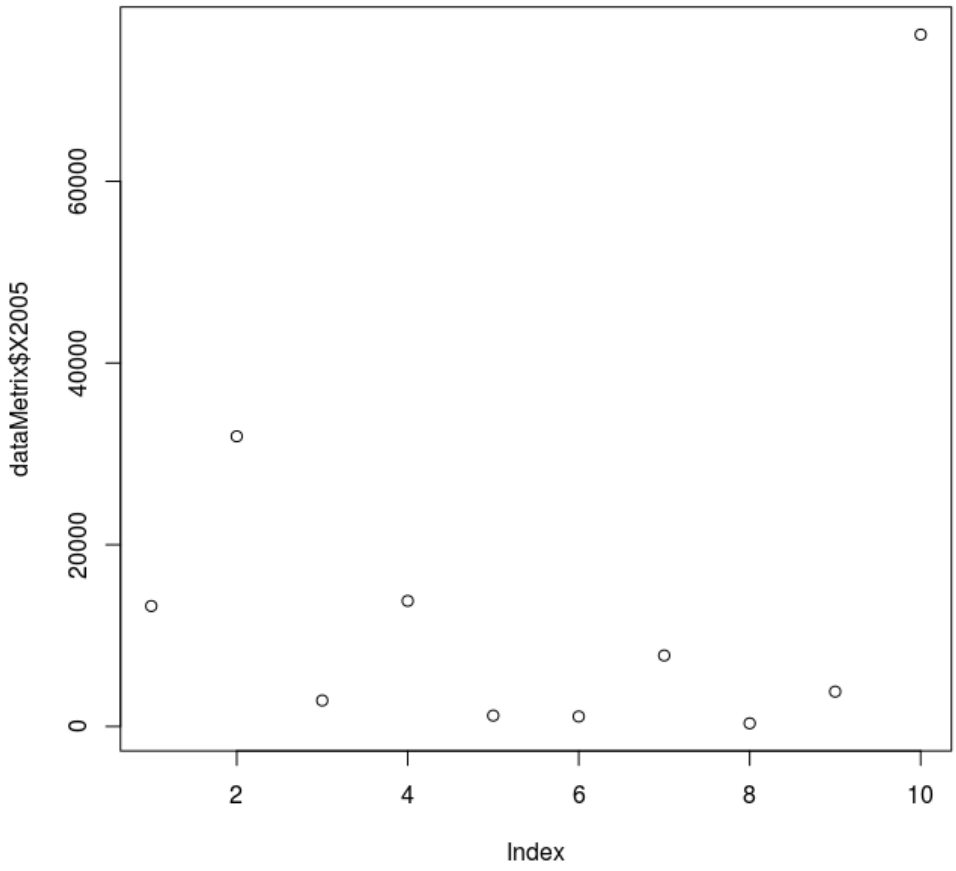
```
jinuser@jin-VirtualBox:~$ sudo chown jinuser:hadoop Ex01.csv
```

```
install.packages("readr")
install.packages("stringi")
library(readr)
library(stringi)
guess_encoding("Ex01.csv")
```

```
> encodingType<-guess_encoding("Ex01.csv");encodingType
# A tibble: 2 x 2
  encoding confidence
     <chr>      <dbl>
1   EUC-KR       1.00
2  GB18030       0.61
```

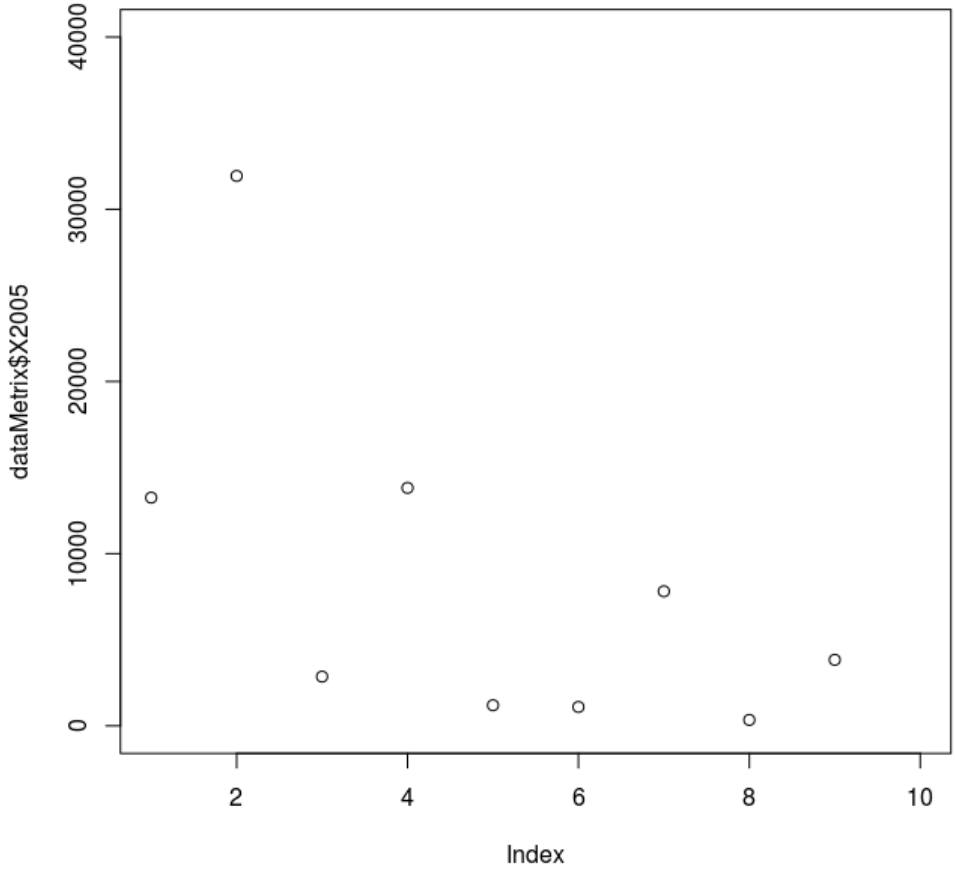
```
> dataMetrix<-read.csv("Ex01.csv", sep = "," , header=TRUE, fileEncoding = "EUC-KR")
> dataMetrix
      IT산업직무별 X2005 X2006 X2007
1           SI개발 13254 13216 15541
2           SW개발 31936 33648 36263
3     디지털콘텐츠  2859  3127  2790
4   시스템운영관리 13819 14554 10759
5  통신/방송서비스  1197  1302   797
6           HW개발  1097  1155  1992
7       HW유지관련  7813  8776  7571
8       IT관련교육   344   419   399
9       IT기술영업  3834  4054  3586
10          합 계 76153 80251 79698
```

```
plot(dataMetrix$X2005)
```



마지막 합계가 존재하여 10번 데이터가 가장 위에 분포하고 있다.
마지막 데이터는 삭제하여 확인할 수도 있지만 다음과 같이 범위를 조절하여 처리할 수도 있다.

```
> plot(dataMetrix$X2005, ylim=c(0, 40000))
```



[참고]r에서 한글문서 열때 인코딩 문제

2017년 10월 21일 토요일 오후 12:38

Philogrammer

r에서 한글문서 열때 인코딩 문제

15 Mar 2017

- Encoding
- Basics

결론부터

R 공부를 시작하고 나서 한글이 들어있는 자료를 분석할 때, 가장 많이 보게되는 에러가 바로 아래와 인코딩 에러입니다.

```
read.csv("http://philogrammer.com/melon10_euc.csv")
## Error in make.names(col.names, unique = TRUE): '<bc><f8><c0><a7>'에서 유효하지 않은 멀티바이트 문자열이 있습니다
```

이것은 바로 R이 설치된 시스템의 인코딩과 실제 우리가 불러들이려는 파일의 인코딩이 맞지 않기 때문에 발생하는 문제입니다.

우리 팀에서 이런 문제가 발생하여 후배들이 많은 고생을 해서 ㅠㅠ 인코딩 에러를 피하기 위하여 read.any 라는 함수를 만들어 보았습니다.

함수를 붙여 넣기 귀찮으신 분들을 위하여 아주 단순한 코드이지만, 패키지화 하여 github에 올려놓았습니다. 아래와 같이 하시면 받으실 수 있습니다.

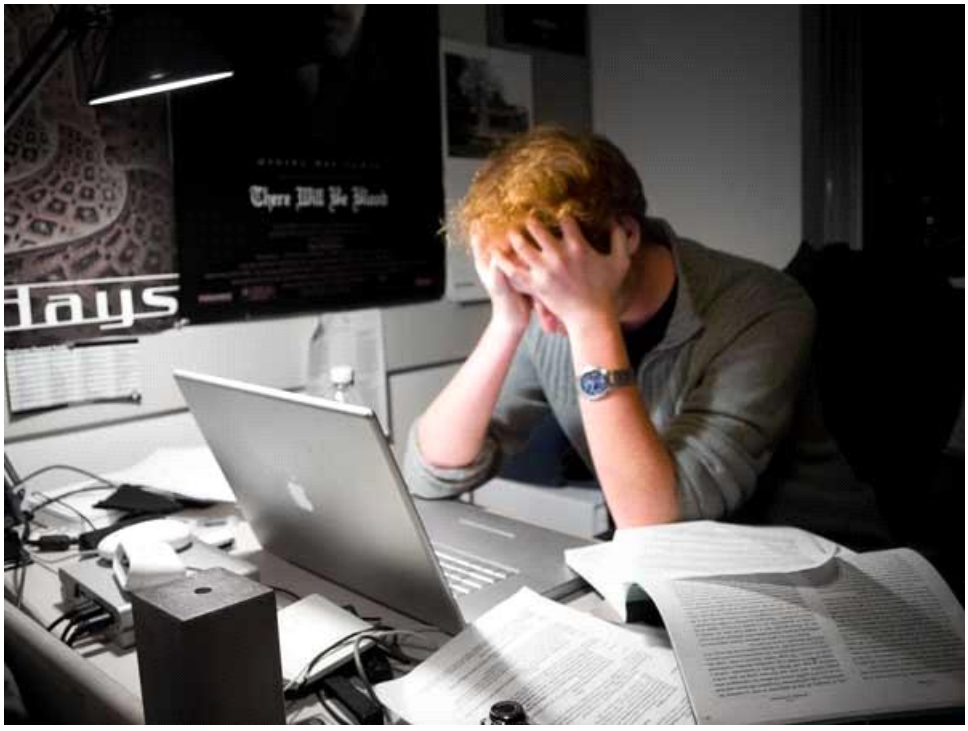
```
# Devtools 패키지를 설치합니다.
library(devtools)
# 패키지를 로드합니다.
install_github("plgrmr/readAny", force = TRUE)
library(readAny)
# 사용법은 read.table 과 100% 똑같습니다.
read.any("http://philogrammer.com/melon10_euc.csv", header = TRUE)
##   순위                          가수
## 1    1                 헤이즈 (Heize)
## 2    2 찬열 (CHANYEOL), 펀치 (Punch)
## 3    3                       정승환
## 4    4                       마마무
## 5    5               TWICE (트와이스)
## 6    6                    지코 (ZICO)
## 7    7                  김희철X민경훈
## 8    8                     BLACKPINK
```

```
## 9     9                세정 (구구단)
## 10   10                볼빨간사춘기
##                          노래제목
## 1                          저 별
## 2                  Stay With Me
## 3                       이 바보야
## 4          Decalcomanie (데칼코마니)
## 5                            TT
## 6  BERMUDA TRIANGLE (Feat. Crush, DEAN)
## 7              나비잠 (Sweet Dream)
## 8                          불장난
## 9              꽃길 (Prod. By ZICO)
## 10                 우주를 줄게
##                                  앨범  좋아요
## 1                                저 별  60,236
## 2                  도깨비 OST Part.1  50,776
## 3                            목소리  55,467
## 4                            MEMORY  57,929
## 5                TWICEcoaster : LANE 1 122,007
## 6                    BERMUDA TRIANGLE  56,163
## 7                나비잠 (Sweet Dream)  73,895
## 8                        SQUARE TWO  61,840
## 9  Jelly box 꽃길 (Prod. By 지코(ZICO)) 세정  60,647
## 10                Full Album RED PLANET 155,328
```

물론 아래 코드를 복사하여 붙여넣기 해서 함수를 생성해서 사용하셔도 무방합니다 :)

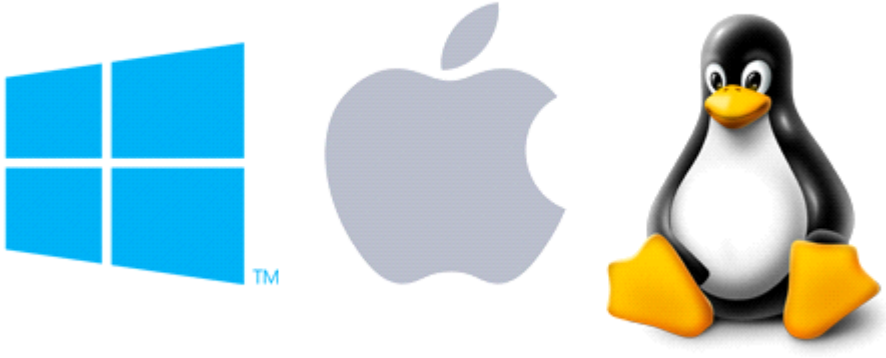
```
library(readr)
read.any <- function(text, sep = "", ...) {
encoding <- as.character(guess_encoding(text)[1,1])
  setting <- as.character(tools::file_ext(text))
  if(sep != "" | !(setting  %in% c("csv", "txt")) ) setting <- "custom"
  separate <- list(csv = ",", txt = "₩n", custom = sep)
  result <- read.table(text, sep = separate[[setting]], fileEncoding = encoding, ...)
  return(result)
}
```

무엇이 문제인가?



R 은 물론이고 Python 이나 SQL 등 다른 여러 언어 등을 시작할 때 가장 먼저 겪게 되는 문제가 바로 인코딩 문제입니다.

excel로 분석에 필요한 데이터를 정리해서 csv로 만든 다음 r에 읽어들였으나 열고나니 외계어가 나오고 영어는 잘 나오는데 한글은 꼬여서 나오기도 하고 이런 경우가 참 많이 있습니다. 사실 경험이 있는 분들에게는 대수로운 것이 아니지만 처음 시작하는 사람 입장에서는 곤욕스럽습니다.



특히 Windows 환경에서 작업하는 작업자와 Mac 상에서 작업하는 작업자 간 csv 로 자료를 교환할 때에 이런 이슈가 자주 발생하곤 합니다.

	EUC-KR	UTF-8
윈도우	EUC-KR 문서를 EUC-KR 환경에서 여는경우	UTF-8 문서를 EUC-KR 환경에서 여는 경우
맥	EUC-KR 문서를 UTF-8 환경에서 여는경우	UTF-8 문서를 UTF-8 환경에서 여는 경우

Mac 에서 작업하여 Export 한 csv 문서는 일반적으로 "UTF-8" 이라고 하는 유니코드 형태로 저장이 됩니다. 반면 Windows 환경에서 Export 된 csv 문서는 "EUC-KR" 인코딩으로 저장이 됩니다.

```
guess_encoding("http://philogrammer.com/melon10_euc.csv")
##    encoding confidence
## 1    EUC-KR       1.00
## 2   GB18030       0.61
## 3      Big5       0.41
```

다른 경우의 수도 있는데요 제가 만들어본 함수에서는 다양한 상황에서 대처 할 수 있도록 문서의 encoding 을 자동으로 추측해서 파일을 읽을 때 지정될 수 있도록 하였습니다.

guess_encoding 함수는 "readr" 패키지를 활용하였습니다.

```
as.character(tools::file_ext("http://philogrammer.com/melon10_euc.csv"))
## [1] "csv"
```

더불어 ",”" 로 분리되어 만들어진 csv 파일과 장문의 txt 파일같은 경우에는 따로 설정하지 않더라도 파일만 넣으면 확장자에 따라 자동으로 구분점을 설정하여 파일을 읽을 수 있게 만들어 놓아 편의를 도왔습니다.

살펴봅시다

```
library(readr)
read.any <- function(text, sep = "", ...) {
# readr 패키지에 있는 guess_encoding 함수를 사용하여
  # 문서의 인코딩을 추측합니다.
  # 다른 언어와 달리 한글과 같은 경우는 높은 확률로
  # 인코딩을 찾아주어 실사용에 큰 무리가 없습니다.
  encoding <- as.character(guess_encoding(text)[1,1])
# 파일 확장자에 따라 구분점을 다르게 처리해주는 부분입니다.
  # 함수에 임의의 인자를 지정해 주는 경우를 제외하고는
  # csv 파일인 경우 "," 를 구분점으로 설정하고
  # txt 파일인 경우 '₩n" 즉 개행문자를 구분점으로 설정하였습니다.
# csv 확장자와 txt 파일인 경우의 대응을 위해 확장자를 setting 변수에 저장해줍니다.
```

```
  setting <- as.character(tools::file_ext(text))
# 임의의 sep 인자를 지정한 경우 혹은 csv나 txt 이외의 확장자를 가진 파일의 경우
  # read.table 함수로 임의의 인자를 넘겨 줄 수 있도록 설정유형을 custom 이라고 해줍니
다.
  if(sep != "" | !(setting  %in% c("csv", "txt")) ) setting <- "custom"
# csv 인 경우 구분자를 , 쉼표 txt 파일인 경우 ₩n 개행문자로 세팅합니다.
  # ₩t 탭 등의 임의의 구분자 지정 시에는 임의의 구분자를 그대로 전달합니다.
  separate <- list(csv = ",", txt = "₩n", custom = sep)
# 결과 값을 전달해줍니다.
  result <- read.table(text, sep = separate[[setting]], fileEncoding = encoding, ...)
return(result)
}
```

Share this:

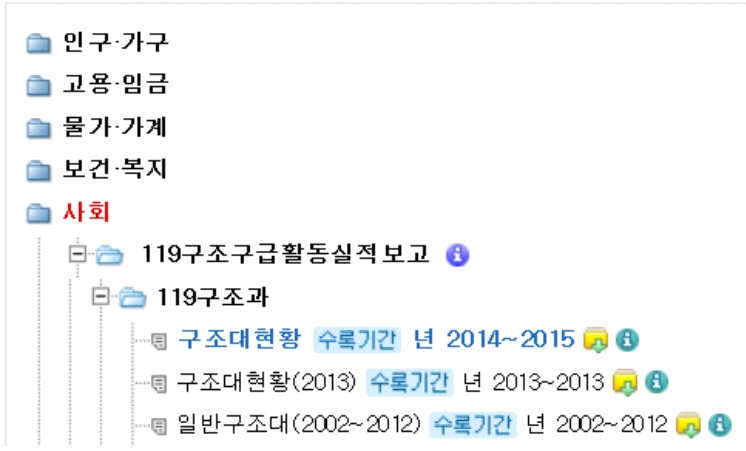
Please enable JavaScript to view the comments powered by Disqus.

<http://philogrammer.com/2017-03-15/encoding/>에서 삽입

히트맵

2017년 10월 21일 토요일 오후 1:14

주제별통계




```
jinuser@jin-VirtualBox:~$ sudo cp /media/sf_Share/구조대현황_20171021132605.csv
./
[sudo] password for jinuser:
jinuser@jin-VirtualBox:~$ sudo mv 구조대현황_20171021132605.csv Ex02.csv
jinuser@jin-VirtualBox:~$ sudo chown jinuser:hadoop Ex02.csv
```

```
> data_Frame<-read.csv("Ex02.csv", sep = "," , header=TRUE, fileEncoding = "EUC-KR")
> data_Frame
      소방본부별        X2015      X2015.1         X2015.2         X2015.3      X2015.4      X2
1     소방본부별          계          계 본부직할특수구조 본부직할특수구조  소방서구조대  소방서구조대
2     소방본부별 조직대수 (대) 편성인원 (명)  조직대수 (대)  편성인원 (명) 조직대수 (대) 편성인원 (명
3           계         267         4189              14            488          212
4          중앙          10          347               4            233            -
5         시·도         257         3851              10            255          212
6     서울특별시          31          633               1             47           23
7     부산광역시          14          292               1             20           11
8     대구광역시          11          174               1             22            9
```

```
> data_Frame[6:24,]
       소방본부별 X2015 X2015.1 X2015.2 X2015.3 X2015.4 X2015.5 X2015.6 X2015.7 X2015.8 X2015.9 X2015.10 X2015.11 X2015.12 X2015.13
6      서울특별시    31     633        1       47       23      489        3       47        3       25         -         -         1        25
7      부산광역시    14     292        1       20       11      239        1       16        -        -         -         -         1        17
8      대구광역시    11     174        1       22        9      137        -        -        -        -         -         -         1        15
9      인천광역시    12     220        1       25        9      162        1       16        -        -         -         -         1        17
10     광주광역시     7     123        1       32        5       80        -        -        -        -         -         -         1        11
11     대전광역시     5      80        -        -        5       80        -        -        -        -         -         -         -         -
```

```
> c(1:15)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
> c(1, 3:10)
[1]  1  3  4  5  6  7  8  9 10
> c(1, seq(from=5, to=15, by=2))
[1]  1  5  7  9 11 13 15
```

```
> data_Frame<-data_Frame[6:24,c(1, seq(from=5, to=15, by=2))]
> data_Frame
       소방본부별 X2015.3 X2015.5 X2015.7 X2015.9 X2015.11 X2015.13
6      서울특별시       47      489       47       25         -        25
7      부산광역시       20      239       16        -         -        17
8      대구광역시       22      137        -        -         -        15
9      인천광역시       25      162       16        -         -        17
10     광주광역시       32       80        -        -         -        11
11     대전광역시        -       80        -        -         -         -
12     울산광역시        -       67        -        -        41        14
```

```
> rownames(data_Frame)<-data_Frame$소방본부별
> data_Frame<-data_Frame[2:7]
> data_Frame
           X2015.3 X2015.5 X2015.7 X2015.9 X2015.11 X2015.13
서울특별시      47      489       47       25         -        25
부산광역시      20      239       16        -         -        17
대구광역시      22      137        -        -         -        15
인천광역시      25      162       16        -         -        17
광주광역시      32       80        -        -         -        11
```

```
c("본부직할특수구조","소방서구조대","수난구조대",산악구조대 화학구조대 항공구조구급대)
```

위의 frame에 헤더를 만들려고 하는데 각 이름에 쌍따옴표를 붙여야한다. 위처럼 블럭설정 후 쌍따옴표를 누르면 앞뒤로 붙게 된다.

```
> colnames(data_Frame)<-c("본부직할특수구조","소방서구조대","수난구조대","산악구조대", "화학구조대", "항공구조구급대")
> data_Frame
           본부직할특수구조 소방서구조대 수난구조대 산악구조대 화학구조대 항공구조구급대
서울특별시                47          489        47        25         -            25
부산광역시                20          239        16         -         -            17
대구광역시                22          137         -         -         -            15
인천광역시                25          162        16         -         -            17
광주광역시                32           80         -         -         -            11
대전광역시                 -           80         -         -         -             -
울산광역시                 -           67         -         -        41            14
세종특별자치시             -            6         -         -         -             -
```

```
data_matrix = data.matrix(data_Frame)
heatmap(data_matrix, Rowv = NA, Colv = NA, col=cm.colors(256), scale = "column", margins = c(10,6))
```

Margins는 hitmap의 여백을 의미하는 것으로 앞의 수는 아래 부분의 여백을 뒤 수는 오른쪽의 여백을 의미함.

히트맵 command

2017년 10월 24일 화요일 오후 4:19

```
system("clear")
data_frame<-read.csv("Ex02.csv", sep=",",fileEncoding="euc-kr")
data_frame[4:24,]

#5번 줄이 6번부터 24번까지의 총 합이므로 제외
c(4, 6:24)
data_frame[c(4, 6:24),1]
data_frame[c(4, 6:24),]

#1번은 이름이 있고 5번부터 15번까지에서 홀 수번째 있는 데이터가 인원에 관한 데이터 이므로 다음과 같이 표기함
c(1, 5, 7, 9, 11, 13, 15)
data_frame[c(4, 6:24),c(1, 5, 7, 9, 11, 13, 15)]

#5,7, 9, 11, 13, 15 너무 없어 보임 따라서 seq를 이용한 홀수 표현을 알아봄
seq(from=1, to=10, by=1)
seq(from=1, to=10, by=2)
seq(from=1, to=10, by=3)
c(1, seq(from=5, to=15, by=2))
data_frame_sub <- data_frame[c(4, 6:24),c(1, seq(from=5, to=15, by=2))]
data_frame_sub

#heatmap을 실행해 보면 수치 행렬이 필요하다고 나옴, 따라서 행렬로 변호나
heatmap(data_frame_sub, col=cm.colors(256))
data_matrix <- data.matrix(data_frame_sub)
heatmap(data_matrix, col=cm.colors(256))

#R console에서 지금까지의 실행 명령어 확인
history()

#row에 이름 지정하기
rownames(data_frame_sub)<-data_frame_sub$소방본부별
data_frame_sub<-data_frame_sub[,2:7]

#col에 이름 지정하기
c("본부직할특수구조", "소방서구조대", "수난구조대", "산악구조대", "화학구조대", "항공구조구급대")
colnames(data_frame_sub)<-c("본부직할특수구조", "소방서구조대", "수난구조대", "산악구조대", "화학구조대", "항공구조구급대")
data_frame_sub
```

```
##############################################################################
##############
#frame을 matrix로 변환
data_matrix<-as.matrix(data_frame_sub)
data_matrix[1]

#"-"를 "0"으로 변환
data_matrix_sub<-gsub("-", "0", data_matrix)
data_matrix_sub[1,]

#matrix를 수치화
data_numeric<-as.numeric(data_matrix_sub)
data_numeric[0:10]

#수치화 된 vector matrix 화
data_test<-matrix(data_numeric, ncol=6)

##############################################################################
##############
#다른 수치화 작업
data_frame_sub<-sapply(data_frame_sub[,1:6],
function(ch){
as.numeric(as.character(ch))
}
)

data_frame_sub[is.na(data_frame_sub)]<-0
data_frame_zero[1:2,]

data_matrix <- data.matrix(data_frame_zero)
```

Map

2017년 10월 21일 토요일 오후 4:06

```
install.packages("maps")
library(maps)

par(mfrow = c(1, 2))
map(database = 'world', region = c('South Korea', 'North Korea'))
title("Korea map in maps packages")
```

par(mfrow = c(1, 2))는 출력 부분을 반으로 나누는 기능을 한다.

Mapdata 활용

```
install.packages("mapproj")
install.packages("mapdata")
```

```
library(mapproj)
library(mapdata)

map(database = 'worldHires', region = c('South Korea', 'North Korea'))
title("Korea map in mapdata packages")



costcos <- read.csv("http://book.flowingdata.com/ch08/geocode/costcos-geocoded.csv", sep=",")
map(database="state")
symbols(costcos$Longitude, costcos$Latitude, circles=rep(1, length(costcos$Longitude)), inches=
0.05, add=TRUE)
```

[참고]Rvis01

2017년 10월 21일 토요일 오후 3:55

Rvis01

Mino

Sunday, January 18, 2015

네이선 아우의 Visualize this 책에 있는 내용 중 R로 실습할 수 있는 내용들을 기준으로 소개하고 일부 내용을 추가하였다

이번 자료에서는 책의 내용들을 위주로 실습하고 이후에는 ggplot2, ggvis 등 다른 시각화 패키지들을 살펴볼것이다

.... 것입니다

우리가 시각화할 자료를 크게 **시간**, **분포**, **관계**, **비교**, **공간** 으로 구분하여

각각의 자료에 맞는 다양한 차트를 R로 그리는 방법을 알아보자

시간 시각화

막대 그래프

일단 데이터를 불러온다

```
hotdogs = read.csv("http://datasets.flowingdata.com/hot-dog-contest-winners.csv",
                   sep = "," ,
                   header = T)
# 이게 정석이다.
# sep옵션을 통해 자료가 무슨 기호를 통해 구분되는지 명시해주고
# header 옵션을 통해 첫번째 열에 열 이름이 있는지 없는지 알려준다
# 근데 없어도 됨
# 개떡같이말해도 찰떡같이 알아듣는다
# 못알아들으면 알아듣게 해줘야 함
head(read.csv("http://datasets.flowingdata.com/hot-dog-contest-winners.csv") )
##   Year                 Winner Dogs.eaten       Country New.record
## 1 1980 Paul Siederman & Joe Baldini      9.10 United States          0
## 2 1981           Thomas DeBerry       11.00 United States          0
## 3 1982            Steven Abrams       11.00 United States          0
## 4 1983              Luis Llamas       19.50        Mexico          0
## 5 1984            Birgit Felden        9.50       Germany          0
## 6 1985          Oscar Rodriguez       11.75 United States          0
```

year 연도

winner 우승자 이름

dogs.eaten 우승자가 먹은 핫도그 수

country 우승자 국적

new.record 세계기록 경신시 1

간단한 막대그래프를 그려보자

barplot 함수를 쓰면 막대그래프를 그릴 수 있다

보통 많이 쓰게 될 plot함수의 경우 기본적으로는 산점도(Scatter Plot)을 그리게 된다

```
barplot(hotdogs$Dogs.eaten)
plot(hotdogs$Dogs.eaten)
```

막대에 연도 라벨 추가

names.arg 옵션을 추가해서 연도에 해당하는 라벨을 만들 수 있다

```
barplot(hotdogs$Dogs.eaten,
        names.arg = hotdogs$Year)
```

축의 라벨, 외곽선 설정, 색상

막대기에 색을 넣어보자

col 에 빨간색을 넣어보고 외곽선은 없애버린다

```
barplot(hotdogs$Dogs.eaten,
        names.arg = hotdogs$Year,
        col = "red",
        border = NA,
        xlab = "Year",
        ylab = "Hotdogs and buns (HDB) eaten")
```

미국인이 우승한 해의 막대와 그렇지 않은 해의 색상을 구분해보자.

색상정보를 담은 리스트 or 벡터를 만들어야 한다

```
fill_colors = c()
for (i in 1:length(hotdogs$Country)){
  if (hotdogs$Country[i] == "United States"){
     fill_colors = c(fill_colors,"#821122")
  } else {
     fill_colors = c(fill_colors, "#cccccc")
  }

}
barplot(hotdogs$Dogs.eaten,
        names.arg = hotdogs$Year,
        col = fill_colors,
        border = NA,
        xlab = "Year",
        ylab = "Hotdogs and buns (HDB) eaten")
```

space 옵션을 통해 막대간격을 조정할 수 있다

main 옵션으로는 제목을 설정한다

```
barplot(hotdogs$Dogs.eaten,
        names.arg = hotdogs$Year,
        col = fill_colors,
        border=NA,
        space = 0.3,
        main = "Nathan's Hot Dog Eating Contest Results, 1980-2010",
        xlab = "Year",
        ylab = "Hotdogs and buns (HDB) eaten")
```

누적 막대 그래프

데이터를 불러온다

```
hot_dog_places = read.csv("http://datasets.flowingdata.com/hot-dog-places.csv",
                  sep = ",",
                  header = T)
head(hot_dog_places)
##   X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007 X2008 X2009 X2010
## 1    25  50.0  50.5  44.5  53.5    49    54    66    59  68.0    54
## 2    24  31.0  26.0  30.5  38.0    37    52    63    59  64.5    43
## 3    22  23.5  25.5  29.5  32.0    32    37    49    42  55.0    37
```

한 열이 한 해의 기록을 나타내고 행은 위에서부터 1,2,3위 입상자를 나타낸다

열이름이 숫자로 되어있을 경우 R에서 자동으로 'X'를 앞에 붙여 String으로 만든다.

원래대로 수정해주자

```
names(hot_dog_places) =
c("2000","2001","2002","2003","2004","2005","2006","2007","2008","2009","2010")
```

이 경우 데이터 프레임인 hot_dog_places 를 행렬로 바꿔줘야한다.

as.matrix 함수를 사용하자

space = 0.25는 막대들 사이의 간격이 막대 너비의 0.25라는 것이다

```
hot_dog_matrix = as.matrix(hot_dog_places)
barplot(hot_dog_matrix,
    border = NA,
    space = 0.25,
    ylim = c(0,200),
    xlab = "Year",
    ylab = "Hot dogs and buns (HDBs) eaten",
    main = "Hot Dog Eating Contest Results, 1980-2010")
```

색을 바꿔보자

```
barplot(hot_dog_matrix,
    border = NA,
    col = c("magenta","cyan","yellow"),
    space = 0.25,
    ylim = c(0,200),
    xlab = "Year",
    ylab = "Hot dogs and buns (HDBs) eaten",
    main = "Hot Dog Eating Contest Results, 1980-2010")
```

스캐터플롯

데이터부터 불러오자

```
subscribers = read.csv("http://datasets.flowingdata.com/flowingdata_subscribers.csv",
                  sep = ",",
                  header = T)
head(subscribers)
##         Date Subscribers Reach Item.Views  Hits
## 1 01-01-2010       25047  4627       9682 27225
## 2 01-02-2010       25204  1676       5434 28042
## 3 01-03-2010       25491  1485       6318 29824
## 4 01-04-2010       26503  6290      17238 48911
## 5 01-05-2010       26654  6544      16224 45521
## 6 01-06-2010       26851  6574      16717 43071
```

Date 날짜

Subscribers 구독자 수

Reach 접속자 수

Item.Views 읽은 글 수

Hits 접속 수

구독자 수로 그림을 그려보자

일단 위에서 설명했듯이 plot으로 그리면 산점도가 나오는게 기본이다

```
plot(subscribers$Subscribers)
```

type 그래프 형태 : p는 점으로 표현되고 h로 하면 고밀도 수직선 그래프

ylim 을 통해 y값의 표현 범위를 정할 수 있다.

```
plot(subscribers$Subscribers,
    type = "p",
    ylim = c(0,30000))
plot(subscribers$Subscribers,
    type = "h",
    ylim = c(0,30000),
    xlab = "Day",
    ylab = "Subscribers")
points(subscribers$Subscribers,
      pch = 19,
      col = "black")
```

plot함수의 경우 실행되면 새로운 그래픽을 생성해버린다

반면 points,lines, abline 등의 함수는 화면에 있는 그래픽위에 내용을 덮어쓰기만 한다

points함수에서 pch 는 점의 크기를 의미한다

```
plot(subscribers$Subscribers, type="h", ylim=c(0,30000), xlab="Day", ylab="Subscribers")
points(subscribers$Subscribers, pch=19, col="black")
abline(0,1000)
```

abline 함수는 abline(a,b) 로 구성되는데 y = a + bx 라고 생각하면 된다

연속형 데이터

시계열 그래프

세계은행에서 발표한 세계 인구 데이터이다

```
population = read.csv("http://datasets.flowingdata.com/world-population.csv",
                      sep = ",",
                      header = T)
head(population)
##   Year Population
## 1 1960 3028654024
## 2 1961 3068356747
## 3 1962 3121963107
## 4 1963 3187471383
## 5 1964 3253112403
## 6 1965 3320396924
```

각각 연도와 인구를 나타낸다

type = "l" 을 이용해 선으로 연결할 수 있다

```
plot(population$Year,population$Population,
    type = "l",
    ylim = c(3000000000, 7000000000),
    xlab = "Year",
    ylab = "Population")
```

계단식 그래프

미국의 우편요금 변화 기록이다

```
postage = read.csv("http://datasets.flowingdata.com/us-postage.csv",
                   sep=",",
                   header=T)
head(postage)
##   Year Price
## 1 1991  0.29
## 2 1995  0.32
## 3 1999  0.33
## 4 2001  0.34
## 5 2002  0.37
## 6 2006  0.39
```

각각 연도와 요금(달러)이다

type = "s"로 계단식 그래프를 그릴 수 있다. s는 step이다 l은 line, p 는 point 겠지..

```
plot(postage$Year, postage$Price,
   type = "s")
plot(postage$Year, postage$Price,
   type = "s",
   main = "US Postage Rates for Letters, First Ounce, 1991-2010",
   xlab = "Year",
   ylab = "Postage Rate (Dallars)")
```

계단식 그래프를 통해 단계적으로 우편 요금이 증가하는 모습을 확인할 수 있다

그냥 선으로 이어버리면 우편요금이 일정하게 유지되는 구간을 보여줄 수 없다

loess

Locally Weighted ScatterPlot Smoothing 으로 데이터의 곡률에 맞는 추세선을 그리는 방법이다

데이터를 작은 조각으로 쪼개서 각 조각마다 추세선을 만들고 종합해서 하나의 곡선 추세선을 형성한다

데이터가 곡선의 추세를 보일 때 사용하자

일단 데이터

미국 실업률 1948-2010

```
unemployment = read.csv("http://datasets.flowingdata.com/unemployment-rate-1948-2010.csv",
                        sep=",")
head(unemployment)
##     Series.id Year Period Value
## 1 LNS14000000 1948    M01   3.4
## 2 LNS14000000 1948    M02   3.8
## 3 LNS14000000 1948    M03   4.0
## 4 LNS14000000 1948    M04   3.9
## 5 LNS14000000 1948    M05   3.5
## 6 LNS14000000 1948    M06   3.6
```

일단 직선으로 fitting한 결과이다

lm 은 linear model 함수인데 여기서는 직선으로 회귀식을 그렸다

intercept와 slope를 반환하기 때문에 abline에 바로 넣으면 직선 추세선을 그릴 수 있다.

```
plot(1:length(unemployment$Value), unemployment$Value)
lnth = 1:length(unemployment$Value)
abline(lm(unemployment$Value ~ lnth))
```

다음은 loess 추세선이다

```
scatter.smooth(x = 1:length(unemployment$Value), y = unemployment$Value)
```

degree와 span으로 곡률을 조정할 수 있다

자세한 옵션이 궁금하면 ?scatter.smooth을 통해 살펴보자

분포 시각화

트리맵

플로잉데이터라는 저자의 블로그에서 가장 인기있는 글 100를 선정하여 카테고리별로 시각화를 하겠다고 한다

```
posts = read.csv("http://datasets.flowingdata.com/post-data.txt")
head(posts)
##     id  views comments            category
## 1 5019 148896       28 Artistic Visualization
## 2 1416  81374       26       Visualization
## 3 1416  81374       26            Featured
## 4 3485  80819       37            Featured
## 5 3485  80819       37             Mapping
## 6 3485  80819       37        Data Sources
```

id 아이디

views 페이지뷰

comments 댓글

category 분류

포트폴리오 패키지는 원래 주식시장의 포트폴리오를 만들기 위해서 쓰는 패키지인데

트리맵 만들기가 편해서 일단 이용해본다

트리맵은 영역 기반의 시각화로, 각 사각형의 넓이가 수치를 나타낸다

위계 구조가 있는 데이터나 트리 구조의 데이터를 표시할 때 적당하다

```
#install.packages("portfolio")
library(portfolio)
## Loading required package: grid
## Loading required package: lattice
## Loading required package: nlme
map.market(id = posts$id,
        area = posts$views,
        group = posts$category,
        color = posts$comments,
        main = "FlowingData Map")
```

id 는 사각형으로 표시할 데이터의 접근자

area는 면적에 반영할 값

group은 사각형을 그룹으로 묶는데 사용되고

color를 이용해 댓글 열에 따라서 구분한다

파이 차트

Visualize This에 내용이 없어서 추가했다

Simple Pie Chart

```
slices <- c(10, 12,4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(slices, labels = lbls, main="Pie Chart of Countries")
```

Pie Chart with Percentages

```
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pct <- round(slices / sum(slices) * 100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,
   labels = lbls,
   col = rainbow(length(lbls)),
   main = "Pie Chart of Countries")
```

Pie Chart from data frame with Appended Sample Sizes

```
mytable <- table(iris$Species)
lbls <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable,
   labels = lbls,
   main = "Pie Chart of Species\n (with sample sizes)")
```

관계 시각화

Scatter Plot

다음 데이터는 미국의 2005년 범죄 유형별 발생건을 인구 100000명 당 발생 비율로 나타낸 결과이다

범죄 유형에 따라 7가지로 나뉜다 (*살인, 절도, 강간, 강도, 폭행, 차량절도, 절취*)

```
crime = read.csv("http://datasets.flowingdata.com/crimeRatesByState2005.csv",
            sep = ",",
            header = T)
head(crime)
##          state murder forcible_rape robbery aggravated_assault burglary
## 1 United States    5.6         31.7   140.7             291.1    726.7
## 2      Alabama    8.2         34.3   141.4             247.8    953.8
## 3       Alaska    4.8         81.1    80.9             465.1    622.5
## 4      Arizona    7.5         33.8   144.4             327.4    948.4
## 5     Arkansas    6.7         42.9    91.1             386.8   1084.6
## 6   California    6.9         26.0   176.1             317.3    693.3
##   larceny_theft motor_vehicle_theft population
## 1        2286.3              416.7  295753151
## 2        2650.0              288.3    4545049
## 3        2599.1              391.0     669488
## 4        2965.2              924.4    5974834
## 5        2711.2              262.1    2776221
## 6        1916.5              712.8   35795255
```

일단 살인과 절도 항목에 대해서만 비교해보자

```
plot(crime$murder, crime$burglary) #살인에 대한 절도 범죄 건수
```

오른쪽 끝의 아웃라이어 때문에 상관관계를 보기가 힘들다

워싱턴DC인데 빼고나서 확인해보자

미국 전체에 대한 항목도 빼버린다

```
crime2 = crime[crime$state != "District of Columbia",] #아웃라이어를 일단 제거하고 해보자
crime2 = crime2[crime2$state != "United States",] #미국 전체 데이터도 제거
plot(crime2$murder, crime2$burglary)
```

축이 0부터 시작하는게 좋은 것 같다

```
plot(crime2$murder, crime2$burglary,
   xlim = c(0,10),
   ylim = c(0,1200))
```

절도와 살인 발생 비율의 관계를 분명하게 확인하기 위해 추세선을 그려보자

```
scatter.smooth(crime2$murder, crime2$burglary,
          xlim = c(0,10),
          ylim = c(0,1200))
```

Scatter Plot matrix

위에서 두 개 변수 사이의 관계를 봤으니 이번에는 여러 변수의 관계를 보고싶다

```
plot(crime2[,2:9])
```

주의 이름을 나타내는 열을 빼고 전부 plot에 넣었더니 이렇게 뽑아준다

추세선을 넣어보자

```
pairs(crime2[,2:9],
    panel = panel.smooth)
```

panel 인수는 x, y에 대한 함수를 변수로 받아 전달한다

예제 코드에선 panel.smooth() 함수(LOESS 추세선을 그려주는 함수)를 전달했다.

자신이 만든 함수도 넣을 수 있다

Burble Chart

스캐터플롯에 버블의 크기로 세 번째 변수를 나타내는 그래프이다

주의해야할 점은 세번째 변수의 값은 반지름이나 지름이 아니라 **면적**으로 표현되어야 한다

```
crime = read.csv("http://datasets.flowingdata.com/crimeRatesByState2005.tsv",
            header = T,
            sep = "\t")
```

앞에 있던 예제와 거의 같은 데이터이다

쉼표 대신 탭으로 구분하는 tsv 파일이므로 구분자를 \t로 변경해주자

```
symbols(crime$murder, crime$burglary,
     circles = crime$population)
```

반지름에 데이터가 들어가니 원이 너무 커진다

원의 면적을 통해 데이터를 표현하자

원의 면적 : pi * r^2

r = sqrt(원의면적/pi) 여기서는 비례를 구하려는 것이니 파이는 무시해도 된다

```
radius = sqrt(crime$population / pi)
symbols(crime$murder, crime$burglary,
     circles = radius) #음.... 원이 전체적으로 다 커보인다
```

모든 원의 크기를 전체적으로 줄일 필요가 있는것같다

```
symbols(crime$murder, crime$burglary,
     circles = radius,
     inches = 0.35,
     fg = "white",
     bg = "red",
     xlab = "Murder Rate",
     ylab = "Burglary Rate")
```

inches : 가장 큰원의 크기를 inch 단위로 설정

fg : foreground

bg : background

symbols 함수는 다른 모형을 선택할 수도 있다.

```
symbols(crime$murder,crime$burglary,
      squares = sqrt(crime$population),
      inches = 0.5)
text(crime$murder, crime$burglary, crime$state,
   cex = 0.5)
```

정사각형 square 직사각형 rectangles 써모미터 thermometers 박스플롯 boxplots 별 stars 등이 선택가능하다

자세한 것은 ?symbols

cex 는 출력 문구의 크기이다 (기본값 1)

히스토그램

데이터는 2002년부터 2009년까지 TV크기 분포 그래프이다

참 신기한 데이터가 많다

```
tvs=read.table("http://datasets.flowingdata.com/tv_sizes.txt",
           sep = "\t",
           header = T)
```

아웃라이어를 제거한다

```
tvs=tvs[tvs$size <80,]
tvs=tvs[tvs$size >10,]
```

히스토그램 구간 수를 설정한다

```
breaks = seq(10, 80, by=5)
```

레이아웃을 설정하고 차트를 그린다

4,2 니깐 4행에 2열 짜리

```
par(mfrow=c(4,2))
hist(tvs[tvs$year == 2009,]$size, breaks=breaks)
hist(tvs[tvs$year == 2008,]$size, breaks=breaks)
hist(tvs[tvs$year == 2007,]$size, breaks=breaks)
hist(tvs[tvs$year == 2006,]$size, breaks=breaks)
hist(tvs[tvs$year == 2005,]$size, breaks=breaks)
hist(tvs[tvs$year == 2004,]$size, breaks=breaks)
hist(tvs[tvs$year == 2003,]$size, breaks=breaks)
hist(tvs[tvs$year == 2002,]$size, breaks=breaks)
```

par(mfrow=c(1,1)) # 이건 그냥 디폴트 설정으로 돌리기 위해서

비교 시각화

히트맵 만들기

2008년의 NBA 농구선수 통계이다

첫번째열은 선수이름, 나머지는 선수 기록 데이터다

```
bball = read.csv("http://datasets.flowingdata.com/ppg2008.csv",
            header = T)
head(bball)
##            Name  G  MIN  PTS  FGM  FGA   FGP FTM FTA   FTP X3PM X3PA
## 1   Dwyane Wade  79 38.6 30.2 10.8 22.0 0.491 7.5 9.8 0.765  1.1  3.5
## 2  LeBron James  81 37.7 28.4  9.7 19.9 0.489 7.3 9.4 0.780  1.6  4.7
## 3   Kobe Bryant  82 36.2 26.8  9.8 20.9 0.467 5.9 6.9 0.856  1.4  4.1
## 4 Dirk Nowitzki  81 37.7 25.9  9.6 20.0 0.479 6.0 6.7 0.890  0.8  2.1
```

```
## 5 Danny Granger  67 36.2 25.8  8.5 19.1 0.447 6.0 6.9 0.878  2.7  6.7
## 6  Kevin Durant  74 39.0 25.3  8.9 18.8 0.476 6.1 7.1 0.863  1.3  3.1
##    X3PP ORB DRB TRB AST STL BLK  TO  PF
## 1 0.317 1.1 3.9 5.0 7.5 2.2 1.3 3.4 2.3
## 2 0.344 1.3 6.3 7.6 7.2 1.7 1.1 3.0 1.7
## 3 0.351 1.1 4.1 5.2 4.9 1.5 0.5 2.6 2.3
## 4 0.359 1.1 7.3 8.4 2.4 0.8 0.8 1.9 2.2
## 5 0.404 0.7 4.4 5.1 2.7 1.0 1.4 2.5 3.1
## 6 0.422 1.0 5.5 6.5 2.8 1.3 0.7 3.0 1.8
```

order 함수를 이용해 원하는 값을 기준으로 정렬한다

행이름을 숫자에서 선수이름으로 바꾸자

```
bball_byfgp = bball[order(bball$FGP, decreasing =T),]
bball = bball[order(bball$PTS, decreasing = F),]
```

row.names(bball) = bball$Name #행이름을 선수 이름으로

```
bball = bball[,2:20]
```

히트맵을 그리려면 데이터프레임이 아니라 행렬 형태여야 한다

```
bball_matrix = data.matrix(bball)
bball_heatmap = heatmap(bball_matrix,
                        Rowv = NA,
                        Colv = NA,
                        col = cm.colors(256),
                        scale = "column",
                        margins = c(5,10))
```

scale 인수를 column으로 설정하면 최대최소값을 전체 행렬이 아니라 열기준으로 결정

cm.colors() 사용할 색상의 범위를 파랑(cyan)부터 빨강(magenta)으로 설정

입력한 숫자 단계(여기서는 256) 길이의 파랑에서 빨강까지의 색상범위를 16진수 색상코드 벡터로 반환

?cm.colors 를 입력하여 자세한 도움말 확인 가능

따뜻한 색감으로 바꿔보자

```
bball_heatmap = heatmap(bball_matrix,
                        Rowv = NA,
                        Colv = NA,
                        col = heat.colors(256),
                        scale = "column",
                        margins = c(5,10))
```

cm.colors(10) #10개의 색상 벡터를 확인할 수 있다.

```
##  [1] "#80FFFFFF" "#99FFFFFF" "#B3FFFFFF" "#CCFFFFFF" "#E6FFFFFF"
##  [6] "#FFE6FFFF" "#FFCCFFFF" "#FFB3FFFF" "#FF99FFFF" "#FF80FFFF"
```

0to255 에서 찾아보는 것도 좋다

색상 코드를 직접 입력해보자

```
red_colors = c("#ffd3cd","#ffc4bc", "#ffb5ab", "#ffa69a", "#ff9789", "#ff8978", "#ff7a67", "#ff6b56",
"#ff5c45", "#ff4d34")
bball_heatmap = heatmap(bball_matrix,
                        Rowv = NA,
                        Colv = NA,
                        col = red_colors,
                        scale = "column",
                        margins = c(5,10))
```

RColorBrewer 패키지

색상 고르기 귀찮으면 설치하자

?brewer.pal 을 통해 자세한 옵션 확인

```
#install.packages("RColorBrewer")
library(RColorBrewer)
bball_heatmap = heatmap(data_matrix, Rowv = NA, Colv = NA, col=brewer.pal(9,"Blues"), scale =
"column", margins = c(10,5))
```

체르노프 페이스

사람의 얼굴을 쉽게 구분한다는 점에서 착안하여, 데이터를 사람의 얼굴 모양에 대입해서 표현한다

유용성은…글쎄

```
#install.packages("aplpack")
library(aplpack)
## Loading required package: tcltk
bball = read.csv("http://datasets.flowingdata.com/ppg2008.csv",
         header = T) #다시 불러오자
faces(bball[,2:16],
    ncolors = 0)
## effect of variables:
##  modified item      Var
##  "height of face   " "G"
##  "width of face    " "MIN"
##  "structure of face" "PTS"
##  "height of mouth  " "FGM"
##  "width of mouth   " "FGA"
##  "smiling          " "FGP"
##  "height of eyes   " "FTM"
##  "width of eyes    " "FTA"
##  "height of hair   " "FTP"
##  "width of hair    "  "X3PM"
##  "style of hair    "  "X3PA"
##  "height of nose   "  "X3PP"
##  "width of nose    "  "ORB"
##  "width of ear     "  "DRB"
##  "height of ear    "  "TRB"
```

faces 함수는 최대 15개의 변수까지만 지원한다

Star Chart

몇 개의 축을 그리고 중앙으로부터의 거리를 통해 수치를 나타낸다

남자들의 경우 위닝에서 선수들 능력치를 본다고 생각하라고 하니까 그림안보고도 이해하더라

```
crime = read.csv("http://datasets.flowingdata.com/crimeRatesByState2005.csv",
         sep = ",",
         header = T) #얘도 다시
stars(crime)
row.names(crime) = crime$state #주 이름을 행이름으로
crime = crime[,2:7] #테이블에서 주 이름을 제거
stars(crime,
    flip.labels = FALSE,
    key.loc = c(15,0.5))
```

flip.labels를 기본값인 T로 놔두면 라벨을 차트의 높이에 따라 위아래를 오가며 배치한다.

key.loc은 범례 위치인것같다

```
stars(crime,
    flip.labels = FALSE,
    key.loc = c(15,1.5),
    full = FALSE)
```

full옵션을 끄면 윗부분만 가지고 그린다

Nightingale chart / Polar Area Diagram

draw.segments 옵션을 켜면 점의 위치 대신 거리로 수치를 나타낸다

```
stars(crime,
    flip.labels = FALSE,
    key.loc = c(15,1.5),
    draw.segments = TRUE)
```

평행좌표계 Parallel coordinates

```
education = read.csv("http://datasets.flowingdata.com/education.csv",
                header=T)
head(education)
##          state reading math writing percent_graduates_sat
## 1 United States     501  515     493                  46
## 2      Alabama     557  552     549                   7
## 3       Alaska     520  516     492                  46
## 4      Arizona     516  521     497                  26
## 5     Arkansas     572  572     556                   5
## 6   California     500  513     498                  49
##   pupil_staff_ratio dropout_rate
## 1              7.9          4.4
## 2              6.7          2.3
## 3              7.9          7.3
## 4             10.4          7.6
## 5              6.8          4.6
## 6             10.9          5.5
```

데이터는 주이름 / SAT 읽기 평균 / 수학평균 / 쓰기평균 / SAT응시비율 / 고등학교졸업직후 취업하는학생비율 / 고교 중퇴율 이다

변수들을 분명한 상관관계로 엮을 수 있을지 알아보자

예를 들면 고교 중퇴율이 높은 주는 과연 평균 SAT점수도 낮을까???

```
library(lattice)
parallelplot(education)
parallelplot(education,
        horizontal.axis = FALSE)  #취향따라...
```

state열은 빼고, 검은색으로 바꿔보자

```
parallelplot(education[,2:7],
        horizontal.axis = FALSE,
        col = "#000000")
```

읽기,수학,쓰기는 평행에 가깝다

SAT점수와 응시율???? 높은 평균점수는 낮은 응시율을 보이고 그 반대도 보인다

읽기 점수를 기준으로 50%씩 나누어서 상위는 검정색, 하위는 회색으로 하자

중앙값은 summary()를 통해 확인

```
summary(education) #reading의 중앙값은 523이다
##       state      reading         math         writing
```

```
##  Alabama   : 1   Min.   :466.0   Min.   :451.0   Min.   :455.0
##  Alaska    : 1   1st Qu.:497.8   1st Qu.:505.8   1st Qu.:490.0
##  Arizona   : 1   Median :523.0   Median :525.5   Median :510.0
##  Arkansas  : 1   Mean   :533.8   Mean   :538.4   Mean   :520.8
##  California: 1   3rd Qu.:571.2   3rd Qu.:571.2   3rd Qu.:557.5
##  Colorado  : 1   Max.   :610.0   Max.   :615.0   Max.   :588.0
##  (Other)   :46
##  percent_graduates_sat pupil_staff_ratio  dropout_rate
##  Min.   : 3.00         Min.   : 4.900     Min.   :-1.000
##  1st Qu.: 6.75         1st Qu.: 6.800     1st Qu.: 2.950
##  Median :34.00         Median : 7.400     Median : 3.950
##  Mean   :37.35         Mean   : 7.729     Mean   : 4.079
##  3rd Qu.:66.25         3rd Qu.: 8.150     3rd Qu.: 5.300
##  Max.   :90.00         Max.   :12.100     Max.   : 7.600
##
reading_colors = c()
for (i in 1:length(education$state)){
 if (education$reading[i] > 523){
    col = "#000000"
 } else {
    col = "#cccccc"
 }
 reading_colors = c(reading_colors, col)
}
parallelplot(education[,2:7],
          horizontal.axis = FALSE,
          col = reading_colors)
```

중퇴율의 경우에는 어떨까

중앙값 대신 첫 4분위수(상위25%)로 필터링 : 여기서는 5.3%

```
dropout_colors = c()
for (i in 1:length(education$state)){
 if (education$dropout_rate[i] > 5.3){
   col = "#000000"
 } else {
   col = "#cccccc"
 }
 dropout_colors = c(dropout_colors, col)
}
parallelplot(education[,2:7],
          horizontal.axis = FALSE,
          col = dropout_colors)
```

……별 의미는 없어보인다

다차원척도법(MDS)

이건 수학적인 설명을 못하겠다

모든 변수를 비교해서 비교하기 가장 좋은 조건을 찾아 2차원에 투영한다고 생각하면 될 것 같다

```
education = read.csv("http://datasets.flowingdata.com/education.csv",
                header = T) #다시
```

ed.dis = dist(education[,2:7]) #모든 주에 대해서 그 주와 나머지 모든 주 사이의 거리를 구한다, 첫열은 주이름이니 제외

#행렬의 한 위치는 행의 주와 열의 주가 얼마만큼의 (유클리드)거리 로 떨어져 있어야 하는지를 보여줌

ed.dis = cmdscale(ed.dis) #cmdscale함수는 거리 행렬을 입력으로 받아서 각 점을 입력된 거리에 맞게 배치한 좌표를 반환한다

```
x = ed.dis[,1] #구한 좌표를 변수에 각각 저장
y = ed.dis[,2]
plot(x,y)
```

한 점이 어떤주를 나타내는지 알 수 없으니 라벨을 붙여보자

```
plot(x,y,
    type = "n") # 이러면 아무것도 안나온다
text(x,y,
    labels = education$state)
```

주 라벨에 dropout_colors를 적용해보자

```
plot(x,y,
    type = "n")
text(x,y,
    labels = education$state,
    col = dropout_colors)
```

별로 의미없어 보인다

```
plot(x,y,
    type = "n")
text(x,y,
    labels = education$state,
    col = reading_colors)
```

오.....

일단 오늘은 여기까지하고

원래 이거 스터디 진행할때는 이거 다음에 바로 지도를 그렸던 것 같은데

일단 자료를 더 모아야 할 것같아서 바로 ggplot2 로 넘어갈 것 같다

근데 ggplot2는 1.0 버전업 이후에 바뀐게 많아서 이것도 고칠게 많을거같음 ㅠㅠ

<http://lumiamitie.github.io/r_tutorial/Rvis01>에서 삽입

```
install.packages("readr")
library(readr)

read.any <- function(text, sep = "", ...) {

  encoding <- as.character(guess_encoding(text)[1,1])
  setting <- as.character(tools::file_ext(text))
  if(sep != "" | !(setting  %in% c("csv", "txt")) )
    setting <- "custom"
  separate <- list(csv = ",", txt = "\n", custom = sep)
  result <- read.table(text, sep = separate[[setting]], fileEncoding = encoding, ...)
  return(result)

}
dataMetrix<-read.any("Ex01.csv", sep = "," , header=TRUE)
dataMetrix
plot(dataMetrix$X2005, ylim=c(0, 40000), type="h")

dataMetrix<-read.csv("Ex02.csv", sep = "," , header=TRUE, fileEncoding = "EUC-KR")
dataMetrix

read.table("Ex01.csv", header = T, encoding = "UTF-8")

encodingType<-guess_encoding("Ex01.csv");encodingType

colnames(dataMetrix)<-dataMetrix[3]
dataMetrix<-dataMetrix[3:24]
dataMetrix[,4:24]
class(dataMetrix)
dataMetrix




data_Frame<-read.csv("Ex02.csv", sep = "," , header=TRUE, fileEncoding = "EUC-KR")
data_Frame
str(dataMetrix)

c(1:15)
c(1, 3:10)
c(1, seq(from=5, to=15, by=2))

data_Frame<-data_Frame[6:24,c(1, seq(from=5, to=15, by=2))]
data_Frame

rownames(data_Frame)<-data_Frame$소방본부별
data_Frame<-data_Frame[2:7]
data_Frame
```

```
c("본부직할특수구조","소방서구조대","수난구조대","산악구조대", "화학구조대", "항공구조구급대")
colnames(data_Frame)<-c("본부직할특수구조","소방서구조대","수난구조대","산악구조대", "화학구조대", "항공구조구급대")
data_Frame

data_matrix = data.matrix(data_Frame)
heatmap(data_matrix, Rowv = NA, Colv = NA, col=cm.colors(256), scale = "column", margins = c(10,6))

library(RColorBrewer)
heatmap(data_matrix, Rowv = NA, Colv = NA, col=brewer.pal(9, "Blues"), margins = c(10,6))

install.packages("maps")
library(maps)

costcos <- read.csv("http://book.flowingdata.com/ch08/geocode/costcos-geocoded.csv", sep=",")
map(database="state")
symbols(costcos$Longitude, costcos$Latitude, circles=rep(1, length(costcos$Longitude)), inches=0.05, add=TRUE)

par(mfrow = c(2, 2))
map(database = 'world', region = c('South Korea', 'North Korea'))
title("Korea map in maps packages")



library(RgoogleMaps)
library(readxl)
seoulwifi <- read_excel("~/rdata/seoulwifi.xlsx")
View(seoulwifi)
map.center.loc <- c(37.5639154,127.0094153)
input.zoom <- 11
map_data <- seoulwifi
win.graph()
mymap <- GetMap(center=map.center.loc, zoom=input.zoom, maptype="road", format="roadmap", destfile="mymap.png")
PlotOnStaticMap(mymap,lat=map_data$yaxis,lon=map_data$xaxis,
destfile="mymap_point.png",cex=1,pch=10,col="red")
```