@RhoMooHyun_bot Twitter

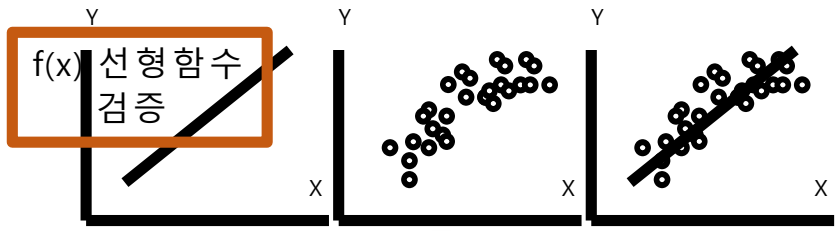
연구방법 in 빅데이터

통계학과 권세혁교수
(http://wolfpack.hnu.ac.kr)

2017.05.24

(1) 통계적 방법론

Confirmatory Data Analysis



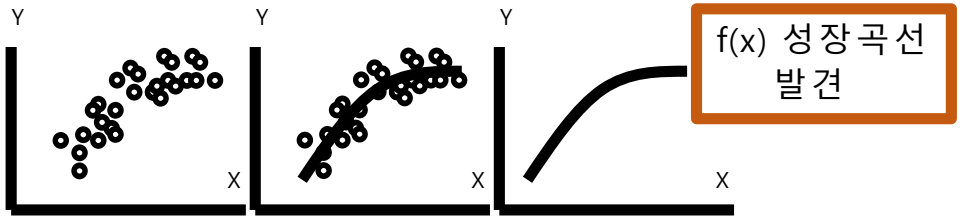
(1) 연구 가설

(2) 데이터 수집

(3) 가설 검정

(4) 결론

Exploratory Data Analysis



(1) 데이터 수집

(2) 데이터 표현

(3) 통계 정보

연구 가설 (CDA)

통계청 MDIS : https://mdis.kostat.go.kr

질병관리본부 국민건강영양조사: http://www.cdc.go.kr/CDC/main.jsp

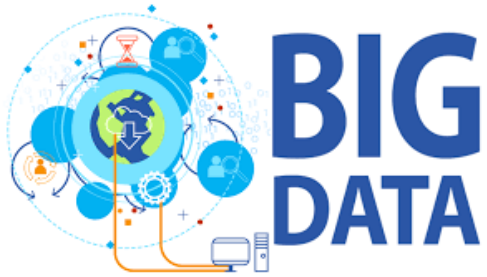
연구조사 / 실험설계

CSV

통계소프트웨어

SAS SPSS EViews

(2) 데이터 수집(전처리 포함) in 빅데이터. – 크롤링 crawling



다양성 Variety
- 텍스트 : 논문, 소설 (PDF, 웹 페이지)
- 사회망 개인 의견 : twitter, Facebook

대용량 Volume
- Garbage in, Information out
- (확증적 연구설계) Knowledge

실시간 Velocity
- 사회경제지표, 주가, 기후, 미세먼지
- Open API : 통계청, 구글, 네이버,…

R
https://www.r-project.org

R Studio
https://www.rstudio.com

python
https://www.python.org

jupyter
http://jupyter.org

(3) 분석 도구 in 빅데이터

http://wolfpack.hnu.ac.kr/Stat_Notes/softwares/software_index.html

감성 분석

- Opinion mining : 텍스트에 나타난 사람들의 태도, 의견, 성향과 같은 주관적 인 데이터를 분석하는 자연어 처리 기술

워드 클라우드

- 문서, 개인의견 사용 단어 빈도 및 시각화

시각화

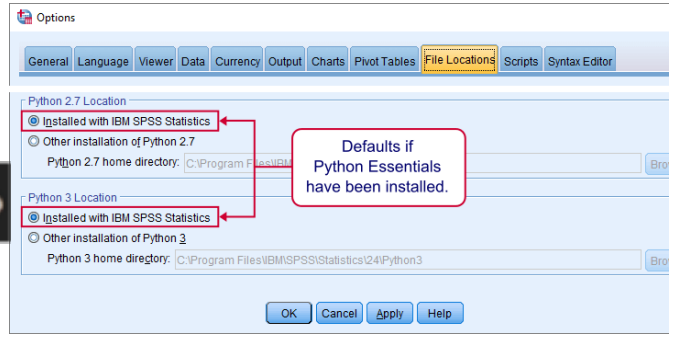
- GGPLOT 라이브러리 함수 – 구글, 네이버 맵

https://www.sas.com/ko_kr/software/university-edition/download-software.html

SAS® University Edition

- Web running SAS + Jupyter Notebook

SPSS Python Essentials - How To Install?



(4) Twitter 워드 클라우드 http://wolfpack.hnu.ac.kr/Big_Data/R_트위터_텍스트마이닝.pdf

계정 만들기

- Twitter 계정 만들기 : https://twitter.com
- twitter Apps Key 만들기 : https://apps.twitter.com

트윗 가져오기

- tweets.usa<-searchTwitter('남북회담',n=1000, lang="ko", since ="2018-05-01", until="2018-05-18", geocode='43.4060924,-77.6977438,1000km')
- tweets.korea<-searchTwitter('남북회담',n=1000, lang="ko", since ="2018-05-01",until="2018-05-18", geocode='37.566535,126.977969,100km')
- tweets.blue<-userTimeline(user='TheBlueHouseKR',n=1000) #노무현대통령 어록 트위터 @RhoMooHyun_bot

데이터 프레임 만들기

- tweets.blue.df<-twListToDF(tweets.blue) 트윗 데이터 포맷 만들기
- tweets.blue.df$text 변수에 내용 저장 -> tweets.text<-tweets.korea.df$text #분석 대상 트윗 내용

한글 자연어 처리

- # 불필요한 문자를 필터링 : tweets.text <- gsub("₩n", "", tweets.text)
- # 문장에서 단어(명사) 분리 - Map 이용 tweets.nouns<-Map(extractNoun, tweets.text)
- #sapply 함수 이용 txt.nouns<-sapply(tweets.text,extractNoun,USE.NAMES = F)
- 자연어처리후 불필요단어제거:gsub()함수

단어빈도 워드 클라우드

- # 단어별 카운팅, 상위 10개 단어 선택 tweets.count<-table(tweets.word)
- wordcloud(names(tweets.count),freq=tweets.count,scale=c(4,0.5),min.freq=5,
- random.order=F,rot.per=.1,colors=pal,family='AppleGothic')

(트윗 검색, 키워드= '남북회담', 2018.05.01~05.17)

서울, 반경 100km, n=1,000

뉴욕, 반경 1,000km, n=1000(236)

(5) PDF영어성경 (word cloud) http://wolfpack.hnu.ac.kr/Big_Data/WORD_CLOUD_영어_PDF.pdf

1) 데이터 읽기

```
install.packages("pdftools")
library(pdftools)
#PDF 문서 txt 데이터 변환
txt <- pdf_text("http://wolfpack.hnu.ac.kr/Big_Data/data/NIV_Bible.pdf")
head(txt,1) #읽은 첫문자 콘솔 출력
```

2) 자연어 처리 – 말뭉치 만들기

```
install.packages(c("NLP","tm"))
library(NLP); library(tm)
docs <- Corpus(VectorSource(txt)) #VectorSource 문자벡터 말뭉치
inspect(docs[2]) #말뭉치 작성 확인 페이지 2
```

3) 특수문자 제거 content_transformer() 함수

```
toSpace<-content_transformer(function (x , pattern ) gsub(pattern,' ', x)) docs <- tm_map(docs, toSpace, '/')
docs <- tm_map(docs, toSpace, '@')
docs <- tm_map(docs, toSpace, '₩₩|')
docs <- tm_map(docs, toSpace, '₩n')
inspect(docs[2]) #특수문자 제외 확인
```

4) 불필요 단어 제거 tm_map() 함수

```
docs <- tm_map(docs, content_transformer(tolower)) # 텍스트 대문자를 소문자로 변환
docs <- tm_map(docs, removeNumbers) # 텍스트 숫자를 제거
docs <- tm_map(docs, removeWords, stopwords("english")) # 텍스트 english 제거
docs <- tm_map(docs, removePunctuation) # 텍스트 물음표 제거
docs <- tm_map(docs, stripWhitespace) # 텍스트 빈칸(white space) 제거
# Text stemming : 동사 기본형으로 표준화 : do, done, doing -> do로 표준화
docs <- tm_map(docs, stemDocument)
# 사용자 지정 제거 단어 설정 및 문서(텍스트)에서 제거
my_custom_stopwords <- c('will', 'come','one', 'said','say','went', 'may', 'let', 'give', 'made','make','came','hous','hand', 'now',
'put','also','call','saw','done','eat','gave','can','left','know','ask','set','even','everi','sent','back','look','took','take')
docs <- tm_map(docs, removeWords, my_custom_stopwords)
inspect(docs[2]) #불필요 단어 제외 확인
```

5) 단어 빈도 계산

```
#문서번호와 단어간의 사용여부 또는 빈도 카운트
dtm <- TermDocumentMatrix(docs)
findFreqTerms(dtm, lowfreq = 500) #500회 이상 사용된 단어 출력
findFreqTerms(dtm, 500,600) #500에서 600회 사이로 사용된 단어 출력
```

```
> findFreqTerms(dtm, lowfreq = 500)
 [1] "day"       "earth"     "god"       "good"      "great"     "heaven"
 [7] "land"      "live"      "place"     "spirit"    "thing"     "two"
[13] "water"     "year"      "eye"       "holi"      "life"      "like"
[19] "lord"      "man"       "name"      "work"      "among"     "answer"
```

```
> findFreqTerms(dtm, 500,600)
 [1] "spirit"    "eye"       "life"      "work"      "answer"    "heard"
 [7] "head"      "destroy"   "righteous" "law"       "christ"
```

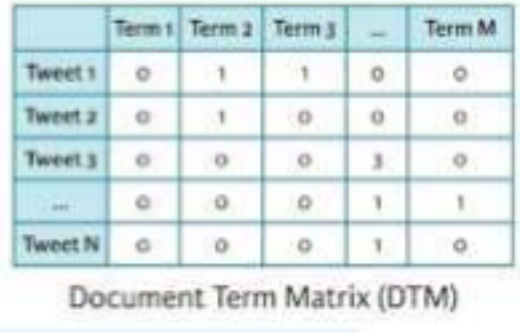
6) 단어 연관분석

```
#findAssocs - 설정 단어(work)와 연관성이 0.3(상관계수) 이상 높은 단어
findAssocs(dtm, terms = "sin", corlimit = 0.3)
findAssocs(dtm, terms = "john", corlimit = 0.3)
```

```
> findAssocs(dtm, terms = "sin", corlimit = 0.3)
$sin
   offer    natur   commit forgiven  sonship
    0.32     0.32     0.31     0.31     0.30

> findAssocs(dtm, terms = "john", corlimit = 0.3)
$john
 baptist  herodia   baptiz  platter    jesus    herod   behead birthday     amaz
    0.53     0.48     0.44     0.42     0.40     0.40     0.38     0.36     0.32
 discipl
    0.32
```

7) DTM 만들기

```
#Document Term matrix 만들기
#The function TermDocumentMatrix() 결과 기반
dtm.mat <- as.matrix(dtm)
v <- sort(rowSums(dtm.mat),decreasing=TRUE)
word.freq <- data.frame(word = names(v),freq=v)
#상위 빈도 20개 빈도표 출력 head(word.freq, 20)
```



	Term 1	Term 2	Term 3	...	Term M
Tweet 1	0	1	1	0	0
Tweet 2	0	1	0	0	0
Tweet 3	0	0	0	3	0
...	0	0	0	1	1
Tweet N	0	0	0	1	0

Document Term Matrix (DTM)

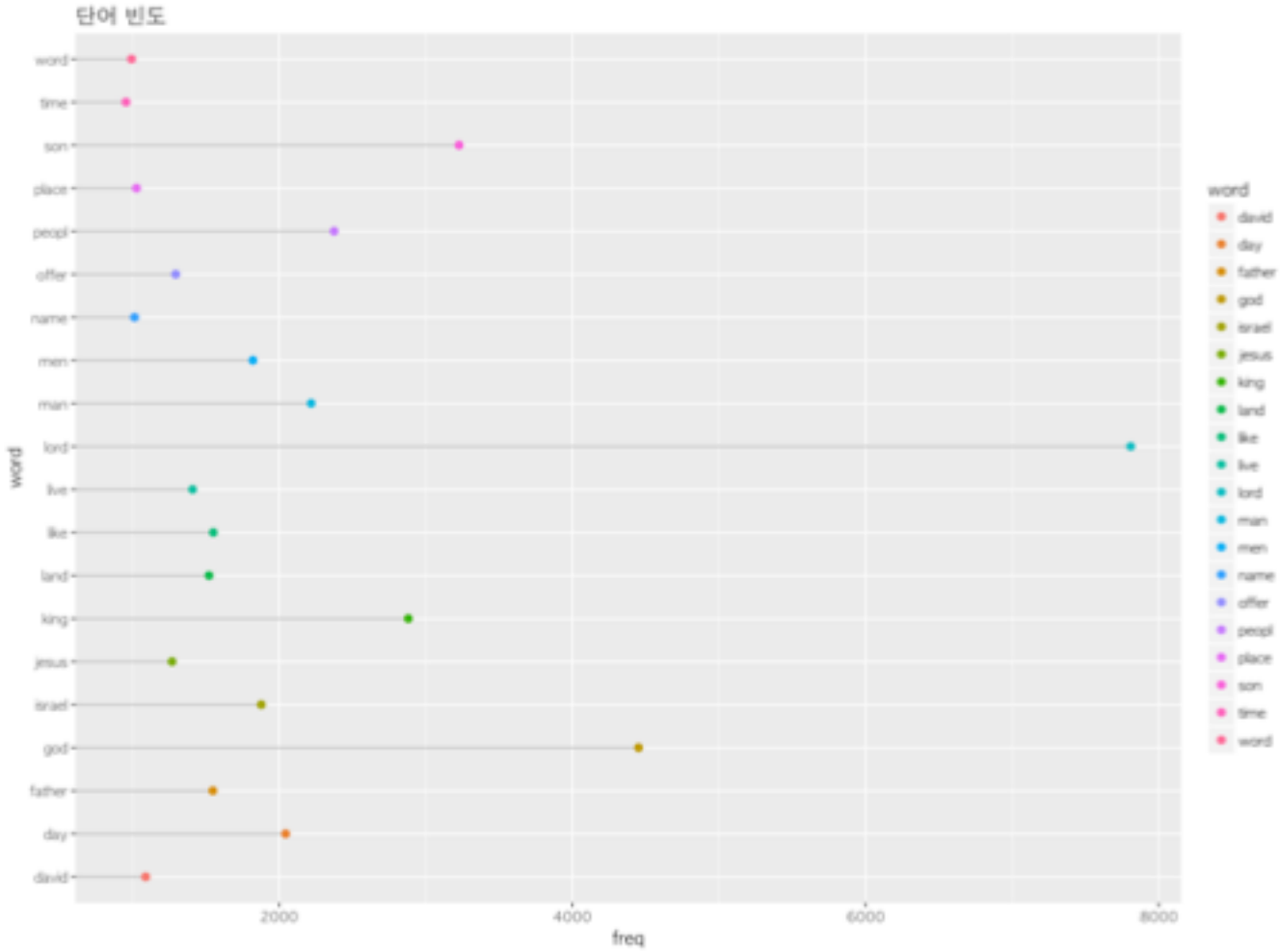
```
> head(word.freq, 20)
          word freq
lord      lord 7809
god        god 4451
son        son 3227
king      king 2881
peopl    peopl 2375
```

```
> head(v,5)
 lord   god   son  king peopl
7809  4451  3227  2881  2375
```

7) 시각화 : 워드 클라우드, 바차트

```
install.packages(c('SnowballC','RColorBrewer','wordcloud'))
library(SnowballC); library(RColorBrewer); library(wordcloud)

#Generate the Word cloud
wordcloud(words = word.freq$word, freq = word.freq$freq, min.freq =
100, random.order=F, rot.per=0.35, colors=brewer.pal(8, 'Dark2'))
```

```
#Frequency Table
require(ggplot2)
ggplot(word.freq[1:20,],aes(x=freq,y=word,col=word)) +
  geom_segment(aes(yend=word),xend=0,color='grey') +
geom_point(size=2,aes(color=word)) +  ggtitle('단어 빈도')
```



한글 PDF 워드 클라우드 : http://wolfpack.hnu.ac.kr/Big_Data/WORD_CLOUD_한글_PDF.pdf

(6) 주가 데이터, 환율, 경제, 미세 데이터, 기상 데이터 가져오기

```
install.packages("quantmod"); library(quantmod)
today <- Sys.Date()

#야후 금융: https://finance.yahoo.com/
kospi <- getSymbols("^KS11",src='yahoo', from='2017-11-01', to='2018-05-23', auto.assign = F)
head(kospi,3); plot(kospi$KS11.Close)
#삼성전자 주가
stock_ss <- getSymbols("005930.KS",src='yahoo', from=today-90, to=today, auto.assign = F)
head(stock_ss,3); plot(stock_ss[,4]) #삼성전자 주가 종가

# OANDA 환율: https://www.oanda.com/
exchange <- getSymbols(Symbols="USD/KRW", src = "oanda", from=today-60, to=today, auto.assign=F)
head(exchange,3)

# 경제 데이터: https://fred.stlouisfed.org/
unemp_xts <- getSymbols(Symbols="LRHUTTTTKRA156N", src = "FRED", from='2017-01-01', to='2018-5-23', auto.assign=F)
head(unemp_xts,3)

# 미세 먼지 데이터 : https://www.airkorea.or.kr/last_amb_hour_data
mise<-read.csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/mise_dj.csv',header=T); head(mise,3)
```

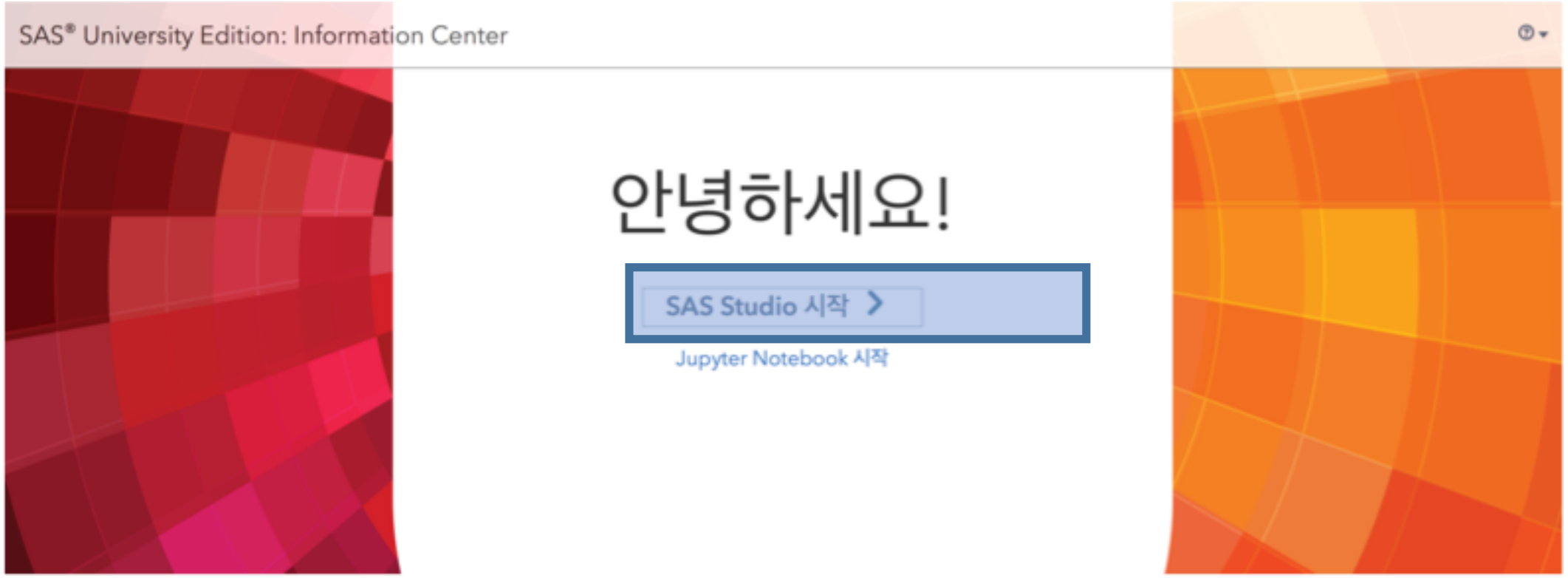
(7) to SAS, SPSS from R

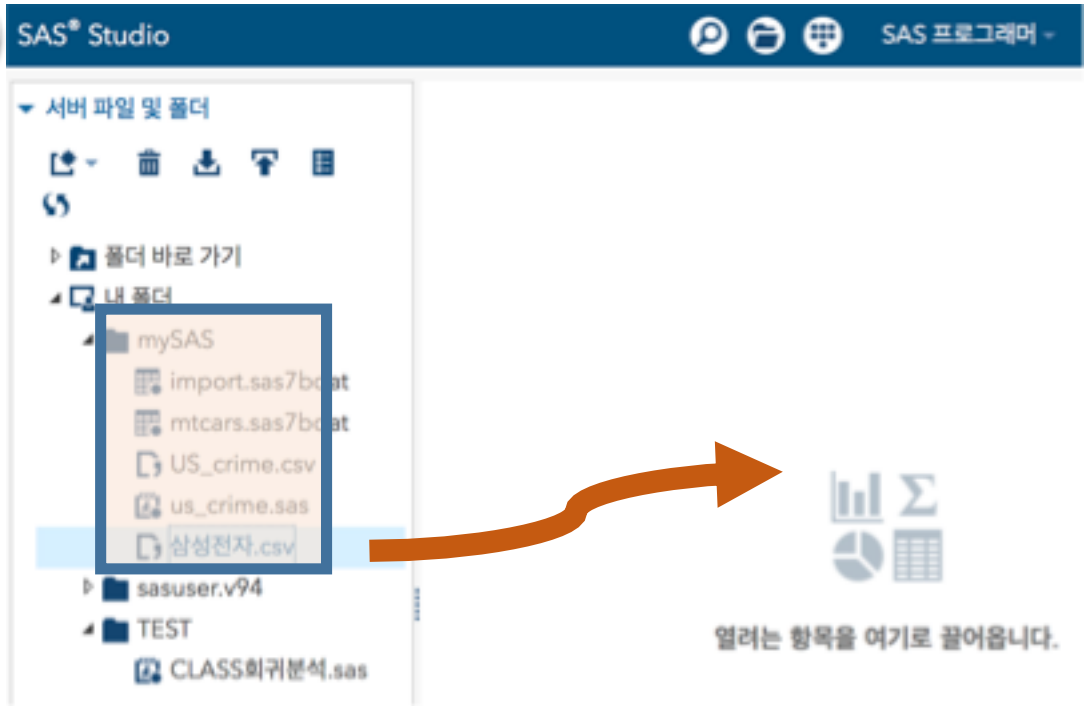
```
#csv format 저장
write.csv(data.frame(stock_ss),"삼성전자.csv") #date 포함

#software data type 저장
install.packages("haven")
library(haven)
data() # R 내장 데이터 리스트
write_sas(mtcars, "mtcars.sas7bdat"). #sas는 변수명이 한글인 경우 작동에 문제 있음
write_sav(mise, "mise.sav")
write_dta(mise, "mise.dta")
```

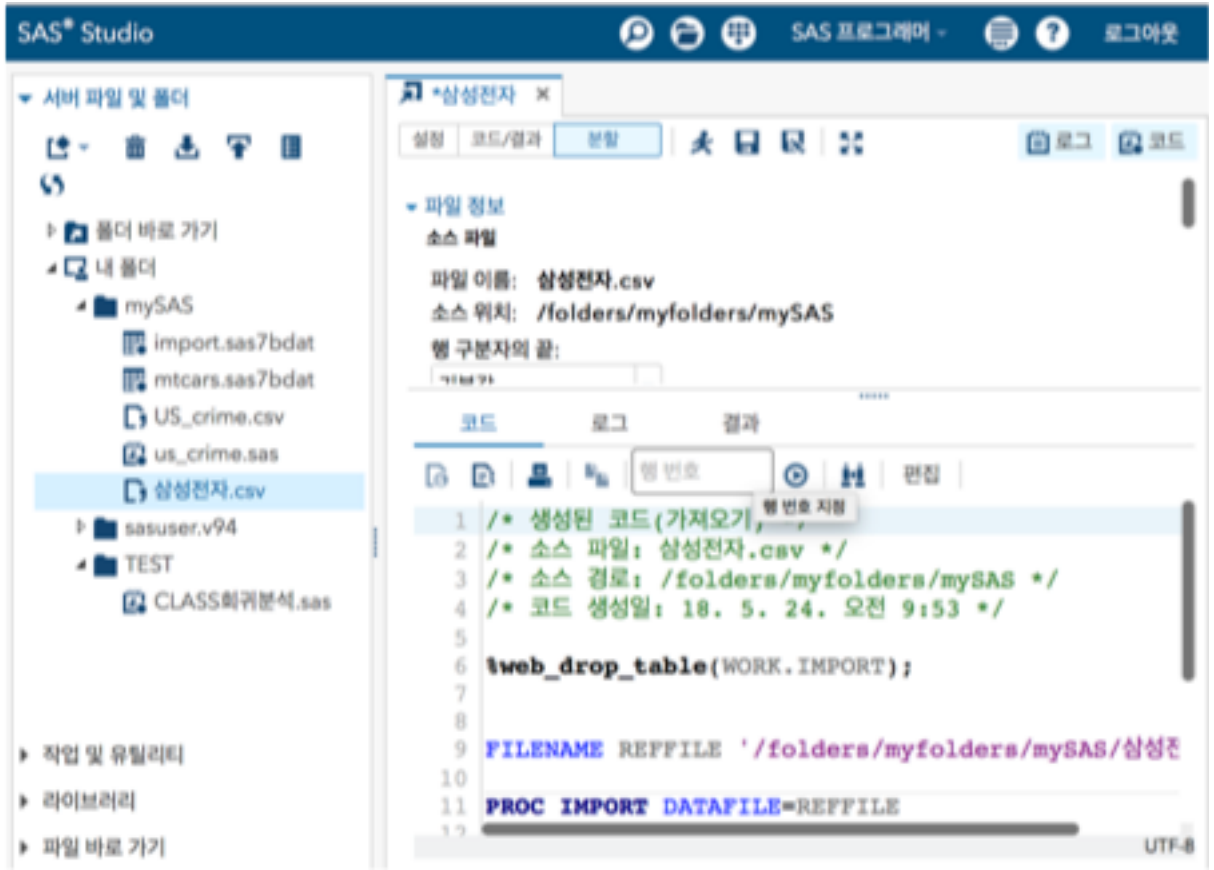
(8) SAS University Edition 사용하기

설치가이드 : https://www.sas.com/ko_kr/software/university-edition/download-software.html

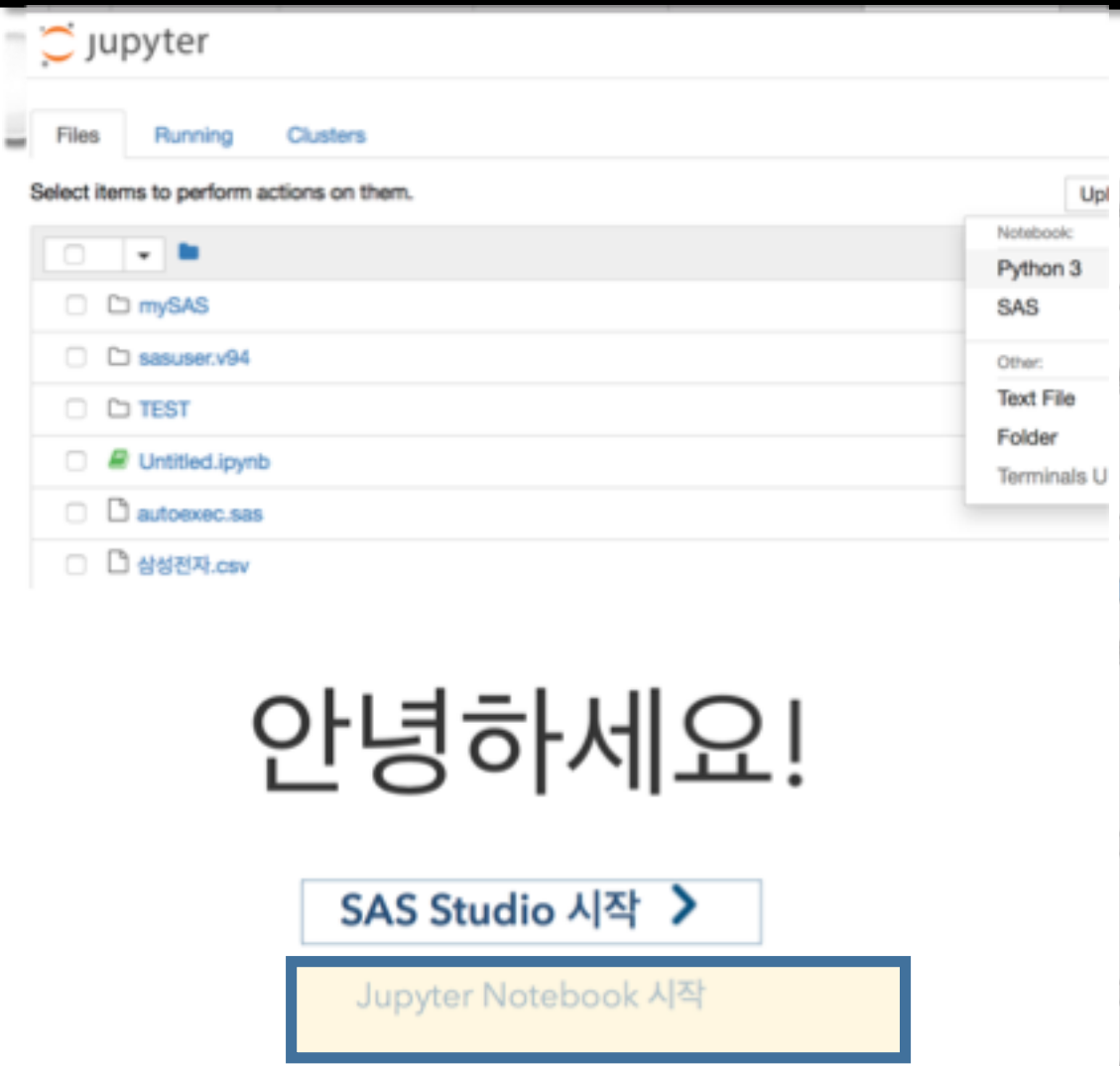
맛보기 : http://wolfpack.hnu.ac.kr/Stat_Notes/softwares/about_R/SAS_Univ.Ed_맛보기.pdf

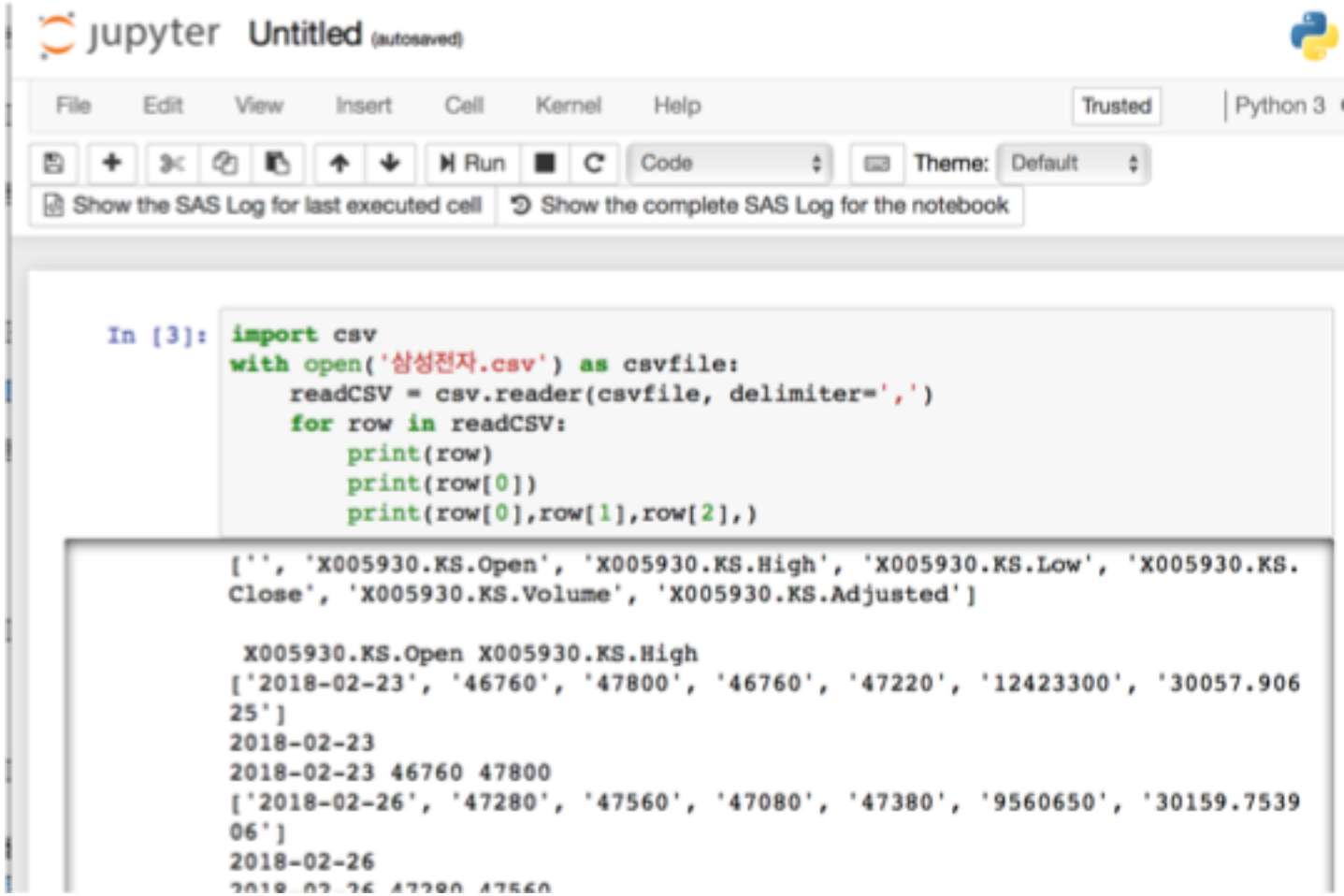



SAS® Studio
SAS 프로그래머
서버 파일 및 폴더
폴더 바로 가기
내 폴더
mySAS
import.sas7bdat
mtcars.sas7bdat
US_crime.csv
us_crime.sas
삼성전자.csv
sasuser.v94
TEST
CLASS회귀분석.sas
열려는 항목을 여기로 끌어옵니다.


SAS® Studio
SAS 프로그래머
로그아웃
서버 파일 및 폴더
폴더 바로 가기
내 폴더
mySAS
import.sas7bdat
mtcars.sas7bdat
US_crime.csv
us_crime.sas
삼성전자.csv
sasuser.v94
TEST
CLASS회귀분석.sas
작업 및 유틸리티
라이브러리
파일 바로 가기
*삼성전자
파일 정보
소스 파일
파일 이름: 삼성전자.csv
소스 위치: /folders/myfolders/mySAS
행 구분자의 끝:
코드
로그
결과
편집
/* 생성된 코드(가져오기,
/* 소스 파일: 삼성전자.csv */
/* 소스 경로: /folders/myfolders/mySAS */
/* 코드 생성일: 18. 5. 24. 오전 9:53 */
%web_drop_table(WORK.IMPORT);
FILENAME REFFILE '/folders/myfolders/mySAS/삼성전
PROC IMPORT DATAFILE=REFFILE
UTF-8

(Python + SAS)


jupyter
Files
Running
Clusters
Select items to perform actions on them.
Upl
Notebook:
Python 3
SAS
Other:
Text File
Folder
Terminals U
mySAS
sasuser.v94
TEST
Untitled.ipynb
autoexec.sas
삼성전자.csv
안녕하세요!
SAS Studio 시작 >
Jupyter Notebook 시작

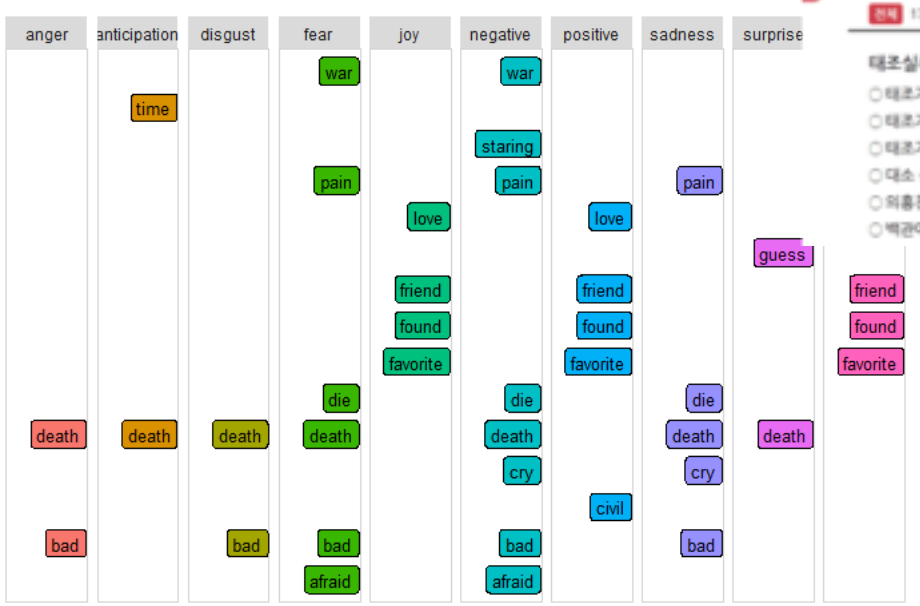

jupyter Untitled (autosaved)
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Theme: Default
Show the SAS Log for last executed cell
Show the complete SAS Log for the notebook
In [3]:
import csv
with open('삼성전자.csv') as csvfile:
readCSV = csv.reader(csvfile, delimiter=',')
for row in readCSV:
print(row)
print(row[0])
print(row[0],row[1],row[2],)
['', 'X005930.KS.Open', 'X005930.KS.High', 'X005930.KS.Low', 'X005930.KS.Close', 'X005930.KS.Volume', 'X005930.KS.Adjusted']
X005930.KS.Open X005930.KS.High
['2018-02-23', '46760', '47800', '46760', '47220', '12423300', '30057.90625']
2018-02-23
2018-02-23 46760 47800
['2018-02-26', '47280', '47560', '47080', '47380', '9560650', '30159.753906']
2018-02-26

1. 오픈 API -java

2. Data Crawling - Python

3. Sentiment analysis

```
> table(get_sentiments('afinn')$score)

 -5  -4  -3  -2  -1   0   1   2   3   4   5
 16  43 264 965 309   1 208 448 172  45   5
> table(get_sentiments('bing')$sentiment) #긍정-부정 이진형

negative positive
    4782     2006
> table(get_sentiments('nrc')$sentiment) #10개의 감성단어

       anger anticipation      disgust         fear          joy
        1247          839         1058         1476          689
    negative     positive      sadness     surprise        trust
        3324         2312         1191          534         1231
```



사회 연결망 분석 (Social Network Analysis : SNA)

개인과 집단들 간의 관계를 노드와 링크로서 모델링해 그것의 위상구조와 확산 및 진화과정을 계량적으로 분석하는 방법론이다.