기상청

[R을 활용한 분석교육실습]

기상 데이터를 활용한 단위 면적 당 양파 생산량 예측 분석 사례

기상청

기상기후
빅데이터 분석 플랫폼

분석 개요

분석 교육 실습 주제인 지상기상관측장비와 양파 재배에 대해 알아봅니다.

● 지상기상관측장비

- 기상청은 서울기상관측소를 비롯하여 전국 102개소의 종관기상관측장비(ASOS)와 무인으로 운영되는 510개소의 자동기상관측장비(AWS)를 이용하여 지상기상관측업무를 수행하고 있다. (2018년 기준) 종관기상 관측장비(ASOS)는 지방청, 기상대, 관측소에 설치되어 기상상태를 관측하고 국제전문 작성 및 통계표 작성과 같은 관측업무를 자동으로 처리한다.
- 관측방법은 기압, 기온, 풍향, 풍속, 습도, 강수량, 강수유무, 일사량, 일조시간, 지면온도, 초상온도, 지중온도, 토양수분, 지하수위 14개 요소에 대해서는 자동으로 관측하고, 시정, 구름, 증발량, 일기 현상 등은 일부 자동과 목측(目測)으로 관측한다.
- AWS는 기상관측소가 없는 곳에 설치되어 집중호우, 우박, 뇌우, 돌풍 등과 같은 국지적인 악기상 현상을 실시간으로 감시하고 있고, 특히 산악지역이나 섬처럼 사람이 관측하기 어려운 곳에 설치되어 집중 호우와 같은 돌발 악기상을 감시하며, 관측된 자료는 수치예보 모델의 초기 입력 자료로 유용하게 사용된다.

● 양파 재배

- 대한민국에서는 8-9월에 모판에 파종하여 10월에 어린 모종을 밭에 정식하고, 다음해 6월 무렵에 수확하는 가을뿌림재 배가 대부분을 차지하고 있다. 봄에 파종하여 가을에 수확하는 봄뿌림 재배를 하면 다음해 1월 상순까지는 싹이 나지 않고, 그 뒤에 냉장하면 4월까지 저장할 수 있다. 봄뿌리재배는 대관령·인제 등지의 고랭지에서 하고 있다. 이 밖에 3-4월에 파종하여 5월 중순경에 작은 알(球)을 수확하고 건조시켰다가 8월 무렵 밭에 심어 겨울부터 이른 봄에 수확하는 세트 재배방식도 있다.
- 양파의 주산지는 2015년 기준 전라남도, 경상남도, 경상북도, 전라북도 등의 순으로 재배면적이 넓게 분포되어 있다.
- 기상으로 인한 양파 생산성 예측모형을 개발하여 양파 생산에 영향을 미치는 기상요소를 파악하고 양파의 생산성 예측을 통해 양파 재배에 도움을 줄 수 있다.

분석 시나리오

실습할 예제는 AWS(자동기상관측장비)와 ASOS(종관기상관측장비)의 기상요소 데이터 및 비기상 데이터를 활용하여 단위 면적 당 양파 생산량을 예측하는 모형을 개발하는것입니다.

● 단위 면적 당 양파 생산량 예측 모형 구축 시나리오

[분석 활용 데이터] → **[Analytics]**

기상 데이터(ASOS)

- (ASOS) 일조시간
- (AWS) 기온, 강수량
- (파생변수)월 평균, 최대, 최소, 합계 기상 요소, 일교차

＋

비기상 데이터

- 양파 생산량
- 양파 재배 면적
- 양파 재배 비율

양파 생산량 영향 변수 선택

정의	회귀계수가 높은 변수를 기준으로 순서대로 상관계수가 0.45 이상인 변수를 제거하여 변수를 선택
분석방법	일반 선형 모델 (GLM ; Generalized linear model), 피어슨 상관분석 (Pearson's Correlation)
분석결과	• 양파 생산량 영향 변수 선택

↓

양파 생산량 산출 모형 구축

정의	일반 선형 모델 (GLM)을 활용하여 단위 면적 당 양파 생산량을 산출하는 산출식 도출
분석방법	일반 선형 모델 (GLM : Generalized linear model)
분석결과	•양파 생산량 산출 모형

분석 절차

실습은 데이터 로딩, 데이터 탐색, 데이터 처리, 모형 구축, 모형 검증의 단계에 따라 진행 될 예정입니다.

● 단위 면적 당 양파 생산량예측 모형 구축 절차

	[실습 설명]	[실습 단계]
1 데이터 로딩	분석환경을 설정하고 분석에 필요한 기상데이터 및 비기상데이터를 로딩하여 분석에 필요한 데이터를 준비하는단계	1. 분석 환경 설정 및 패키지로딩 2. 데이터불러오기 3. 데이터결합하기
2 데이터 탐색	분석 데이터의 데이터 타입을변환하고 탐색적 자료 분석(Exploratory Data Analysis)을 수행하는 단계	1. 타입 변환 2. 탐색적 자료 분석
3 데이터 처리	이상치와 결측치를 대체하며 파생변수를 생성하여 최종 분석 데이터셋을 완성하는 단계	1. 이상치 및 결측치 대체 2. 파생변수 생성 3. 분석 데이터셋 완성
4 모형 구축	데이터간의 다중공선성을 제거하여 단위면적 당 양파 생산량 영향 변수를 선택하는 단계	1. 다중공선성 해결 및 변수 선택 2. 모형 구축
5 모형 검증	최종 구축한 산출 모형의 성능을 검증하는 단계	1. 교차검증

분석 데이터

단위 면적 당 양파 생산량 예측 모형 구축에 사용된 파일 및 변수 정보를 확인합니다.

● 단위 면적 당 양파 생산량 예측 모형 구축에 사용된 파일 및 변수 정보

파일명	파일 설명	변수명	변수 설명	형식	예제
AWS_TA_DD	자동기상 관측자료 (일 기온)	TMA	날짜 (연도/월/일)	chr	2000-01-01
		STN_ID	관측기기 번호	chr	129
		AVG_TA	일 평균 기온	num	6.1
		MAX_TA	일 최대 기온	num	21.4
		MIN_TA	일 최소 기온	num	0.6
		YM	날짜 (연도/월)	chr	2000-01
AWS_RN_DD	자동기상 관측자료 (일 강수)	TMA	날짜 (연도/월/일)	chr	2000-01-01
		STN_ID	관측기기 번호	chr	243
		SUM_RN	일 합계 강수량	num	0
		YM	날짜 (연도/월)	chr	2000-01
ASOS_SS_DD	종관기상 관측자료 (일조시간)	TMA	날짜 (연도/월/일)	chr	2000-01-01
		STN_ID	관측기기 번호	chr	90
		SUM_SS_HR	일 합계 일조시간	num	0
		YM	날짜 (연도/월)	chr	2000-01
onion	양파 생산량	region_1	지역 (시도)	chr	경남
		year	연도	int	2001
		y	당해 생산량	int	6045
onion.area	양파 재배면적	STN_ID	관측기기 번호	int	285
		region_1	지역 (시도)	chr	경남
		region_2	지역 (시군구)	chr	합천
		area_id	지역 번호	chr	4800000000
		area	양파 재배면적	int	1205
		w	면적 비율	num	0.403

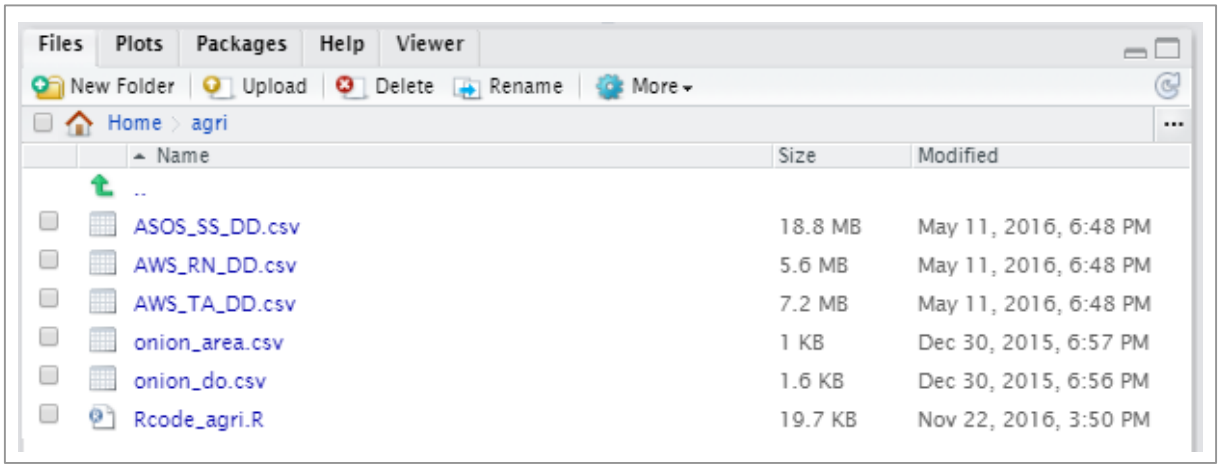
기상청

Ⅰ. 데이터 로딩

1. 분석 환경 설정 및 패키지 로딩
2. 데이터 불러오기
3. 데이터 결합

1. 분석 환경 설정 및 패키지 로딩 (1/2)

- **교육실습 R 소스**
  - 분석을 실행하기 위한 예제 소스 위치
    . ./agri/Rcode_agri.R



- **옵션 세팅**
  - 분석을 실행하기 전 메모리를 초기화 하고 옵션들을 지정

```
#=================================================================#
#Setting global option
#=================================================================#
rm(list=ls()) #remove objects
gc() #garbage collection
options(scipen=100, digits=5) #deciding to print : 100, digits to print : 5
Sys.setenv(LANG = "en") #To English
#=================================================================#
```

- rm() 로 모든 할당 된 객체들을 삭제
- gc() 로 메모리에 쌓여 있는 불필요한 데이터 정리
- options() 로 소수점 위 100자리까지, 소수점 아래 5자리까지의 옵션을세팅
- Sys.setnev() 로 에러 메시지를 영어로 출력하게 세팅

1. 분석 환경 설정 및 패키지 로딩 (2/2)

- **패키지 로딩 및 H2O 세팅**

  - 분석을 위해 설치 된 패키지를 로딩 및 H2O를 세팅 함

```
#=================================================================#
#Loading library
#=================================================================#
library(data.table); require(ggplot2); library(plyr); library(reshape)
library(reshape2); library(stringr); library(h2o)
.libPaths()

#Create an H2O cloud
h2o.init(ip = "localhost", port = 54321)
h2o.removeAll() #Clean slate just in case the cluster was already running
#=================================================================#
```

  - **library() 로 설치 된 패키지를 로딩**
    - data.table, plyr, reshape, reshape2, stringr : 데이터 핸들링을 위한 라이브러리
    - ggplot2 : 그림을 그리기 위한 라이브러리
    - h2o : 데이터 분석을 위한 라이브러리
  - **.libPaths() 로 할당 된 패키지 경로를 확인**
  - **h2o.init() 로 H2O를 세팅**

2. 데이터 불러오기(1/3)

- **경로 설정**

- 데이터를 로딩 할 기본 경로를 확인

```
#=================================================================#
#Setting Path
#=================================================================#
getwd()
#=================================================================#
```

  - **getwd() 로 경로 확인**

- **데이터 로딩**

-분석을 위해 필요한 데이터를 로딩 함

```
##AWS_TA_DD
AWS_TA_DD <-fread("./agri/AWS_TA_DD.csv")
AWS_TA_DD <-subset(AWS_TA_DD, select= c("tma", "stn_id", "avg_ta", "max_ta", "min_ta"))
colnames(AWS_TA_DD) <- c("TMA", "STN_ID", "AVG_TA", "MAX_TA", "MIN_TA")

AWS_TA_DD[, TMA := substr(TMA, 1,10)]
AWS_TA_DD[, STN_ID := as.character(STN_ID)]
AWS_TA_DD$YM <- substr(AWS_TA_DD$TMA, 1, 7)
str(AWS_TA_DD)

##AWS_RN_DD
AWS_RN_DD <- fread("./agri/AWS_RN_DD.csv")
AWS_RN_DD <- subset(AWS_RN_DD, select=c(TMA, STN_ID, SUM_RN))
colnames(AWS_RN_DD) <- c("TMA", "STN_ID", "SUM_RN")

AWS_RN_DD[, SUM_RN := as.numeric(SUM_RN)]
AWS_RN_DD[, TMA := substr(TMA, 1,10)]
AWS_RN_DD[, STN_ID := as.character(STN_ID)]
AWS_RN_DD$YM <- substr(AWS_RN_DD$TMA, 1, 7)
str(AWS_RN_DD)

##ASOS_SS_DD
ASOS_SS_DD <- fread("./agri/ASOS_SS_DD.csv")
ASOS_SS_DD <- subset(ASOS_SS_DD, select=c(TMA, STN_ID, SUM_SS_HR))
colnames(ASOS_SS_DD) <- c("TMA", "STN_ID", "SUM_SS_HR")

ASOS_SS_DD[, TMA := substr(TMA, 1,10)]
ASOS_SS_DD[, STN_ID := as.character(STN_ID)]
ASOS_SS_DD$YM <-substr(ASOS_SS_DD$TMA, 1, 7)
str(ASOS_SS_DD)
```

2. 데이터 불러오기(2/3)

```
##onion
onion <- fread("./agri/onion_do.csv", encoding="UTF-8“, stringsAsFactors=FALSE)
names(onion) <-c("region_1", "year", "y")
str(onion)

##onion.area
onion <- fread("./agri/onion_area.csv", encoding="UTF-8", stringsAsFactors=FALSE,
integer64="character")
str(onion.area)
#=================================================================#
```

- **fread() 로 데이터 로딩**
- **colnames(), names() 로 컬럼명 변경**
- **str() 로 데이터 속성 및 형태 확인**
- **substr() 로 텍스트 일부분 추출**
- **as.character() 로 데이터 형태를 문자 형식으로 변환**
- **subset() 로 데이터 일부분 추출**

> 실행 결과

```
> #=================================================================#
> # Loading Data
> #=================================================================#
> ## AWS_TA_DD
> AWS_TA_DD <- fread("./agri/AWS_TA_DD.csv")
> AWS_TA_DD <- subset(AWS_TA_DD, select= c("tma", "stn_id", "avg_ta", "max_ta", "min_ta"))
> colnames(AWS_TA_DD) <- c("TMA", "STN_ID", "AVG_TA", "MAX_TA", "MIN_TA")
>
> AWS_TA_DD[, TMA := substr(TMA, 1, 10)]
> AWS_TA_DD[, STN_ID := as.character(STN_ID)]
> AWS_TA_DD$YM <- substr(AWS_TA_DD$TMA, 1, 7)
> str(AWS_TA_DD)
Classes ‘data.table’ and 'data.frame':  151944 obs. of  6 variables:
 $ TMA   : chr  "2000-01-01" "2000-01-01" "2000-01-01" "2000-01-01" ...
 $ STN_ID: chr  "129" "243" "278" "281" ...
 $ AVG_TA: num  6.1 3.5 -0.3 2.2 7 6.3 8.1 7.3 3 0.3 ...
 $ MAX_TA: num  21.4 9.4 5.2 6.4 10.5 12.8 13.9 12.7 7.7 5.6 ...
 $ MIN_TA: num  0.6 -1.6 -7.1 -4 1.9 -0.7 2.6 0.8 -2.5 -6.6 ...
 $ YM    : chr  "2000-01" "2000-01" "2000-01" "2000-01" "2000-01" ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> ## AWS_RN_DD
> AWS_RN_DD <- fread("./agri/AWS_RN_DD.csv")
> AWS_RN_DD <- subset(AWS_RN_DD, select=c(TMA, STN_ID, SUM_RN))
> colnames(AWS_RN_DD) <- c("TMA", "STN_ID", "SUM_RN")
>
> AWS_RN_DD[, SUM_RN := as.numeric(SUM_RN)]
> AWS_RN_DD[, TMA := substr(TMA, 1, 10)]
> AWS_RN_DD[, STN_ID := as.character(STN_ID)]
```

2. 데이터 불러오기(3/3)

```
> AWS_RN_DD$YM <- substr(AWS_RN_DD$TMA, 1, 7)
> str(AWS_RN_DD)
Classes 'data.table' and 'data.frame':  151944 obs. of  4 variables:
 $ TMA   : chr  "2000-01-01" "2000-01-01" "2000-01-01" "2000-01-01" ...
 $ STN_ID: chr  "243" "260" "261" "281" ...
 $ SUM_RN: num  0 0 0 0 0 0 0 0 0 0 ...
 $ YM    : chr  "2000-01" "2000-01" "2000-01" "2000-01" ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> ## ASOS_SS_DD
> ASOS_SS_DD <- fread("./agri/ASOS_SS_DD.csv")
> ASOS_SS_DD <- subset(ASOS_SS_DD, select=c(TMA, STN_ID, SUM_SS_HR))
> colnames(ASOS_SS_DD) <- c("TMA", "STN_ID", "SUM_SS_HR")
>
> ASOS_SS_DD[, TMA := substr(TMA, 1, 10)]
> ASOS_SS_DD[, STN_ID := as.character(STN_ID)]
> ASOS_SS_DD$YM <- substr(ASOS_SS_DD$TMA, 1, 7)
> str(ASOS_SS_DD)
Classes 'data.table' and 'data.frame':  486970 obs. of  4 variables:
 $ TMA      : chr  "2000-01-01" "2000-01-01" "2000-01-01" "2000-01-01" ...
 $ STN_ID   : chr  "90" "95" "98" "100" ...
 $ SUM_SS_HR: num  0 3.8 4.3 0 5.1 0 0 2.3 4.6 3.3 ...
 $ YM       : chr  "2000-01" "2000-01" "2000-01" "2000-01" ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> ## onion
> onion <- fread("./agri/onion_do.csv", encoding="UTF-8", stringsAsFactors=FALSE)
> names(onion) <- c("region_1", "year", "y")
> str(onion)
Classes 'data.table' and 'data.frame':  90 obs. of  3 variables:
 $ region_1: chr  "경남" "경남" "경남" "경남" ...
 $ year    : int  2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 ...
 $ y       : int  6045 6150 5883 6538 6365 6198 7057 6995 8386 7153 ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> ## onion.area
> onion.area <- fread("./agri/onion_area.csv",  encoding="UTF-8", stringsAsFactors=FALSE,
+                     integer64="character")
> str(onion.area)
Classes 'data.table' and 'data.frame':  26 obs. of  6 variables:
 $ STN_ID  : int  285 295 912 919 278 281 812 813 822 823 ...
 $ region_1: chr  "경남" "경남" "경남" "경남" ...
 $ region_2: chr  "합천" "남해" "함양" "창녕" ...
 $ area_id : chr  "4800000000" "4800000000" "4800000000" "4800000000" ...
 $ area    : int  1205 NA 665 1121 83 179 509 97 537 204 ...
 $ w       : num  0.403 0 0.222 0.375 0.052 0.111 0.316 0.06 0.334 0.127 ...
 - attr(*, ".internal.selfref")=<externalptr>
> #================================================================#
```

3. 데이터결합

- **데이터 결합**

- 두 데이터를 한 데이터로 결합

```
#=============================================================#
#Merging Data
#=============================================================#
##AWS_TA_DD and AWS_RN_DD
AWS_TA_RN <-merge(x=AWS_TA_DD, y=AWS_RN_DD, by=c("STN_ID", "TMA", "YM"),
all=TRUE)
str(AWS_TA_RN); rm(AWS_TA_DD, AWS_RN_DD)

##AWS_TA_RN and ASOS_SS_DD
TA_RN_SS <-merge(x=AWS_TA_RN, y=ASOS_SS_DD, by=c("STN_ID", "TMA", "YM"),
                 all.x=TRUE, all.y=FALSE)
str(TA_RN_SS); rm(AWS_TA_RN, ASOS_SS_DD)
#=============================================================#
```

- **merge() 로 두 데이터를 결합**

> 실행 결과

```
> #=============================================================#
> # Merging Data
> #=============================================================#
> ## AWS_TA_DD and AWS_RN_DD
> AWS_TA_RN <- merge(x=AWS_TA_DD, y=AWS_RN_DD, by=c("STN_ID", "TMA", "YM"), all=TRUE)
> str(AWS_TA_RN); rm(AWS_TA_DD, AWS_RN_DD)
Classes 'data.table' and 'data.frame':  151944 obs. of  7 variables:
 $ STN_ID: chr  "129" "129" "129" "129" ...
 $ TMA   : chr  "2000-01-01" "2000-01-02" "2000-01-03" "2000-01-04" ...
 $ YM    : chr  "2000-01" "2000-01" "2000-01" "2000-01" ...
 $ AVG_TA: num  6.1 5.4 -0.4 1.1 4.1 3.9 -5.8 -2 -1 0.2 ...
 $ MAX_TA: num  21.4 8.3 2.9 5.4 7.3 8.1 -0.4 2.3 0.9 3.1 ...
 $ MIN_TA: num  0.6 -0.5 -3.6 -4 1.2 -1.6 -8.6 -5.7 -5.3 -3.7 ...
 $ SUM_RN: num  0 4 0 0 34 7.5 1 0.5 0 0 ...
 - attr(*, ".internal.selfref")=<externalptr>
 - attr(*, "sorted")= chr  "STN_ID" "TMA" "YM"
>
> ## AWS_TA_RN and ASOS_SS_DD
> TA_RN_SS <- merge(x=AWS_TA_RN, y=ASOS_SS_DD, by=c("STN_ID", "TMA", "YM"), all.x=TRUE, all.y=FAL
> str(TA_RN_SS); rm(AWS_TA_RN, ASOS_SS_DD)
Classes 'data.table' and 'data.frame':  151944 obs. of  8 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ TMA      : chr  "2000-01-01" "2000-01-02" "2000-01-03" "2000-01-04" ...
 $ YM       : chr  "2000-01" "2000-01" "2000-01" "2000-01" ...
 $ AVG_TA   : num  6.1 5.4 -0.4 1.1 4.1 3.9 -5.8 -2 -1 0.2 ...
 $ MAX_TA   : num  21.4 8.3 2.9 5.4 7.3 8.1 -0.4 2.3 0.9 3.1 ...
 $ MIN_TA   : num  0.6 -0.5 -3.6 -4 1.2 -1.6 -8.6 -5.7 -5.3 -3.7 ...
 $ SUM_RN   : num  0 4 0 0 34 7.5 1 0.5 0 0 ...
 $ SUM_SS_HR: num  4.3 2.2 7.5 5.2 0 0.2 7.4 0.8 0 7.2 ...
 - attr(*, ".internal.selfref")=<externalptr>
 - attr(*, "sorted")= chr  "STN_ID" "TMA" "YM"
> #=============================================================#
```

Ⅱ. 데이터 탐색

1. 타입 변환
2. 탐색적 자료 분석

1. 타입변환

- **타입 변환**

- 분석을 실행하기 위해 변수들의 타입을 재정의

```
#=================================================================#
#Changing Type
#=================================================================#
ls()
##TA_RN_SS
TA_RN_SS[, STN_ID := as.factor(STN_ID)]
TA_RN_SS[, TMA :=as.character(TMA)]
TA_RN_SS[, YM := as.factor(YM)]
str(TA_RN_SS)

##onion
onion[, region_1 :=as.factor(region_1)]
str(onion)

##onion.area
onion.area[, STN_ID := as.factor(STN_ID)]
onion.area[, region_1 :=as.factor(region_1)]
onion.area[, region_2 :=as.factor(region_2)]
onion.area[, area_id :=as.factor(area_id)]
str(onion.area)
#=================================================================#
```

- **ls() 로 현재 R 메모리에 존재하는 데이터확인**
- **as.factor() 로 타입을 요소로 전환**
- **as.character() 로 타입을 문자로 전환**

2. 탐색적 자료 분석(1/9)

- **탐색적 자료 분석 (Exploratory Data Analysis ; EDA)**

- 분석을 실행하기 위해 데이터 및 변수의 형태를 확인

```
#============================================================#
#Exploratory Data Analysis
#============================================================#
##TA_RN_SS
#Summary
summary(TA_RN_SS); str(TA_RN_SS)

summary(subset(TA_RN_SS, is.na(AVG_TA))); str(subset(TA_RN_SS, is.na(AVG_TA)))
summary(subset(TA_RN_SS, is.na(MAX_TA))); str(subset(TA_RN_SS, is.na(MAX_TA)))
summary(subset(TA_RN_SS, is.na(MIN_TA))); str(subset(TA_RN_SS, is.na(MIN_TA)))

summary(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)));
str(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)))
head(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)),20)

summary(subset(TA_RN_SS, is.na(SUM_RN))); str(subset(TA_RN_SS, is.na(SUM_RN)))
summary(subset(TA_RN_SS, is.na(SUM_SS_HR))); str(subset(TA_RN_SS,  is.na(SUM_SS_HR)))

#Boxplot
par(mfrow=c(3,1))
boxplot(data=TA_RN_SS, AVG_TA~STN_ID, xlab="STN_ID", ylab="AVG_TA",
                main="Boxplot of AVG_TA and STN_ID")
boxplot(data=TA_RN_SS, MAX_TA~STN_ID, xlab="STN_ID", ylab="MAX_TA",
                main="Boxplot of MAX_TA and STN_ID")
boxplot(data=TA_RN_SS, MIN_TA~STN_ID, xlab="STN_ID", ylab="MIN_TA",
                main="Boxplot of MIN_TA andSTN_ID")

par(mfrow=c(1,1))
boxplot(data=TA_RN_SS, SUM_SS_HR~STN_ID, xlab="STN_ID", ylab="SUM_SS_HR",
                main="Boxplot of SUM_SS_HR and STN_ID")

##onion
summary(onion)

boxplot(data=onion, y~region_1, xlab="region_1", ylab="y", main="Boxplot of y and
region_1" )
boxplot(data=onion, y~year, xlab="year", ylab="y", main="Boxplot of y and year" )
```

2. 탐색적 자료 분석(2/9)

```
##onion.area
summary(onion.area)

boxplot(data=onion.area, area~region_1, xlab="region_1", ylab="area",
                main="Boxplot of area andregion_1")

qplot(x=w, y=area, data=subset(onion.area, !is.na(w) & !is.na(area)), color=region_1,
                geom = c("path", "jitter"))
#=================================================================#
```

- summary() 로 데이터의 요약 통계량출력
- head() 로 데이터의 헤더 정보출력
- par() 로 그래프 출력 형태 설정
- boxplot() 로 상자 그림 출력
- qplot() 로 그래프 출력

> 실행 결과

```
> #=================================================================#
> # Exploratory Data Analysis
> #=================================================================#
> ## TA_RN_SS
> # Summary
> summary(TA_RN_SS); str(TA_RN_SS)
     STN_ID           TMA                 YM             AVG_TA
 129    :  5844   Length:151944      2000-01:   806   Min.   :-14.4
 170    :  5844   Class :character   2000-03:   806   1st Qu.:  5.2
 184    :  5844   Mode  :character   2000-05:   806   Median : 14.2
 189    :  5844                      2000-07:   806   Mean   : 13.4
 243    :  5844                      2000-08:   806   3rd Qu.: 21.7
 260    :  5844                      2000-10:   806   Max.   : 31.7
 (Other):116880                      (Other):147108   NA's   :438
     MAX_TA          MIN_TA           SUM_RN          SUM_SS_HR
 Min.   :-8.8   Min.   :-27.80   Min.   :  0.00   Min.   : 0
 1st Qu.:11.0   1st Qu.: -0.20   1st Qu.:  0.00   1st Qu.: 2
 Median :20.4   Median :  8.60   Median :  0.00   Median : 6
 Mean   :19.0   Mean   :  8.54   Mean   :  3.71   Mean   : 6
 3rd Qu.:26.9   3rd Qu.: 17.70   3rd Qu.:  0.50   3rd Qu.: 9
 Max.   :40.4   Max.   : 28.80   Max.   :420.00   Max.   :14
 NA's   :29     NA's   :19       NA's   :30       NA's   :82457
Classes 'data.table' and 'data.frame':  151944 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ TMA      : chr  "2000-01-01" "2000-01-02" "2000-01-03" "2000-01-04" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ AVG_TA   : num  6.1 5.4 -0.4 1.1 4.1 3.9 -5.8 -2 -1 0.2 ...
 $ MAX_TA   : num  21.4 8.3 2.9 5.4 7.3 8.1 -0.4 2.3 0.9 3.1 ...
 $ MIN_TA   : num  0.6 -0.5 -3.6 -4 1.2 -1.6 -8.6 -5.7 -5.3 -3.7 ...
 $ SUM_RN   : num  0 4 0 0 34 7.5 1 0.5 0 0 ...
 $ SUM_SS_HR: num  4.3 2.2 7.5 5.2 0 0.2 7.4 0.8 0 7.2 ...
 - attr(*, ".internal.selfref")=<externalptr>
>
```

2. 탐색적 자료 분석(3/9)

```
> summary(subset(TA_RN_SS, is.na(AVG_TA))); str(subset(TA_RN_SS, is.na(AVG_TA)))
     STN_ID         TMA                  YM          AVG_TA         MAX_TA
 912    : 42   Length:438         2001-06: 13   Min.   : NA   Min.   :-3.2
 733    : 35   Class :character   2000-09: 11   1st Qu.: NA   1st Qu.:11.8
 714    : 33   Mode  :character   2000-12: 11   Median : NA   Median :21.4
 732    : 32                      2002-02: 10   Mean   :NaN   Mean   :19.7
 734    : 25                      2000-04:  9   3rd Qu.: NA   3rd Qu.:27.4
 919    : 25                      2002-08:  9   Max.   : NA   Max.   :38.8
 (Other):246                      (Other):375   NA's   :438   NA's   :18
     MIN_TA           SUM_RN         SUM_SS_HR
 Min.   :-19.60   Min.   :  0.00   Min.   : 0.00
 1st Qu.:  0.50   1st Qu.:  0.00   1st Qu.: 2.50
 Median : 12.40   Median :  0.00   Median : 6.70
 Mean   :  9.96   Mean   :  5.74   Mean   : 5.96
 3rd Qu.: 19.70   3rd Qu.:  0.50   3rd Qu.: 8.85
 Max.   : 27.00   Max.   :393.00   Max.   :12.60
 NA's   :17       NA's   :16       NA's   :307
Classes 'data.table' and 'data.frame':  438 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ TMA      : chr  "2000-01-19" "2000-01-25" "2000-01-26" "2000-01-27" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 1 1 1 3 4 9 19 33 39 ...
 $ AVG_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MAX_TA   : num  0 -2.5 NA -2.2 7 19.5 26 30.7 18.7 5.5 ...
 $ MIN_TA   : num  -4.7 -6.7 NA -10.3 -1.6 6.1 13.7 23.5 14.7 -3.1 ...
 $ SUM_RN   : num  3 0 NA 1.5 0.5 0 0 66 4 0 ...
 $ SUM_SS_HR: num  0 8.2 4.4 7.9 5.1 10.5 4.5 2.7 7.7 10.2 ...
 - attr(*, ".internal.selfref")=<externalptr>
> summary(subset(TA_RN_SS, is.na(MAX_TA))); str(subset(TA_RN_SS, is.na(MAX_TA)))
     STN_ID       TMA                  YM          AVG_TA         MAX_TA
 732    : 4   Length:29          2015-06:10   Min.   :10.6   Min.   : NA
 129    : 3   Class :character   2001-08: 2   1st Qu.:14.6   1st Qu.: NA
 733    : 3   Mode  :character   2001-10: 2   Median :15.0   Median : NA
 281    : 2                      2002-02: 2   Mean   :15.0   Mean   :NaN
 714    : 2                      2003-04: 2   3rd Qu.:15.4   3rd Qu.: NA
 734    : 2                      2012-06: 2   Max.   :20.3   Max.   : NA
 (Other):13                      (Other): 9   NA's   :18     NA's   :29
     MIN_TA          SUM_RN        SUM_SS_HR
 Min.   : 3.1   Min.   : 0.00   Min.   :0.00
 1st Qu.:10.3   1st Qu.: 0.75   1st Qu.:0.00
 Median :11.5   Median : 4.00   Median :1.10
 Mean   :12.0   Mean   : 4.07   Mean   :3.69
 3rd Qu.:13.4   3rd Qu.: 5.75   3rd Qu.:7.90
 Max.   :19.7   Max.   :12.50   Max.   :9.40
 NA's   :17     NA's   :15      NA's   :18
Classes 'data.table' and 'data.frame':  29 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 3 4 6 9 10 10 11 ...
 $ TMA      : chr  "2000-01-26" "2003-04-12" "2012-06-01" "2012-06-01" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 40 150 150 186 188 186 186 191 186 ...
 $ AVG_TA   : num  NA NA NA NA 20.3 NA 14.5 15.5 10.6 15.1 ...
 $ MAX_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MIN_TA   : num  NA NA NA NA 19.7 17.7 10.2 13.3 3.1 11.8 ...
 $ SUM_RN   : num  NA NA NA NA 7.5 0 4.5 4 0 6 ...
 $ SUM_SS_HR: num  4.4 0 8.4 9.4 0 7.4 0 0 9.2 1.1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

2. 탐색적 자료 분석(4/9)

```
> summary(subset(TA_RN_SS, is.na(MIN_TA))); str(subset(TA_RN_SS, is.na(MIN_TA)))
     STN_ID       TMA                  YM         AVG_TA          MAX_TA
 732    :4   Length:19          2001-08:2   Min.   :11.5   Min.   :12.7
 129    :3   Class :character   2001-10:2   1st Qu.:13.4   1st Qu.:15.6
 733    :3   Mode  :character   2002-02:2   Median :15.2   Median :18.4
 714    :2                      2003-04:2   Mean   :15.2   Mean   :18.4
 734    :2                      2012-06:2   3rd Qu.:17.1   3rd Qu.:21.2
 184    :1                      2012-11:2   Max.   :19.0   Max.   :24.1
 (Other):4                      (Other):7   NA's   :17     NA's   :17
     MIN_TA         SUM_RN        SUM_SS_HR
 Min.   : NA   Min.   : 0.00   Min.   :0.00
 1st Qu.: NA   1st Qu.: 0.00   1st Qu.:0.45
 Median : NA   Median : 0.00   Median :3.10
 Mean   :NaN   Mean   : 7.88   Mean   :4.00
 3rd Qu.: NA   3rd Qu.: 7.88   3rd Qu.:7.40
 Max.   : NA   Max.   :31.50   Max.   :9.40
 NA's   :19    NA's   :15      NA's   :13
Classes 'data.table' and 'data.frame':  19 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 3 5 7 13 14 14 15 ...
 $ TMA      : chr  "2000-01-26" "2003-04-12" "2012-06-01" "2012-06-01" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 40 150 150 189 192 40 155 155 20 ..
 $ AVG_TA   : num  NA NA NA NA 19 11.5 NA NA NA NA ...
 $ MAX_TA   : num  NA NA NA NA 24.1 12.7 NA NA NA NA ...
 $ MIN_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SUM_RN   : num  NA NA NA NA 0 31.5 NA 0 0 NA ...
 $ SUM_SS_HR: num  4.4 0 8.4 9.4 1.8 0 NA NA NA NA ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> summary(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)));
     STN_ID       TMA                  YM         AVG_TA          MAX_TA
 732    :4   Length:17          2001-08:2   Min.   : NA   Min.   : NA
 129    :3   Class :character   2001-10:2   1st Qu.: NA   1st Qu.: NA
 733    :3   Mode  :character   2002-02:2   Median : NA   Median : NA
 714    :2                      2003-04:2   Mean   :NaN   Mean   :NaN
 734    :2                      2012-06:2   3rd Qu.: NA   3rd Qu.: NA
 184    :1                      2012-11:2   Max.   : NA   Max.   : NA
 (Other):2                      (Other):5   NA's   :17    NA's   :17
     MIN_TA         SUM_RN      SUM_SS_HR
 Min.   : NA   Min.   :0    Min.   :0.00
 1st Qu.: NA   1st Qu.:0    1st Qu.:3.30
 Median : NA   Median :0    Median :6.40
 Mean   :NaN   Mean   :0    Mean   :5.55
 3rd Qu.: NA   3rd Qu.:0    3rd Qu.:8.65
 Max.   : NA   Max.   :0    Max.   :9.40
 NA's   :17    NA's   :15   NA's   :13
> str(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)))
Classes 'data.table' and 'data.frame':  17 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 3 13 14 14 15 15 15 ...
 $ TMA      : chr  "2000-01-26" "2003-04-12" "2012-06-01" "2012-06-01" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 40 150 150 40 155 155 20 20 23 ...
 $ AVG_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MAX_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MIN_TA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SUM_RN   : num  NA NA NA NA NA 0 0 NA NA NA ...
 $ SUM_SS_HR: num  4.4 0 8.4 9.4 NA NA NA NA NA NA ...
 - attr(*, ".internal.selfref")=<externalptr>
```

2. 탐색적 자료 분석(5/9)

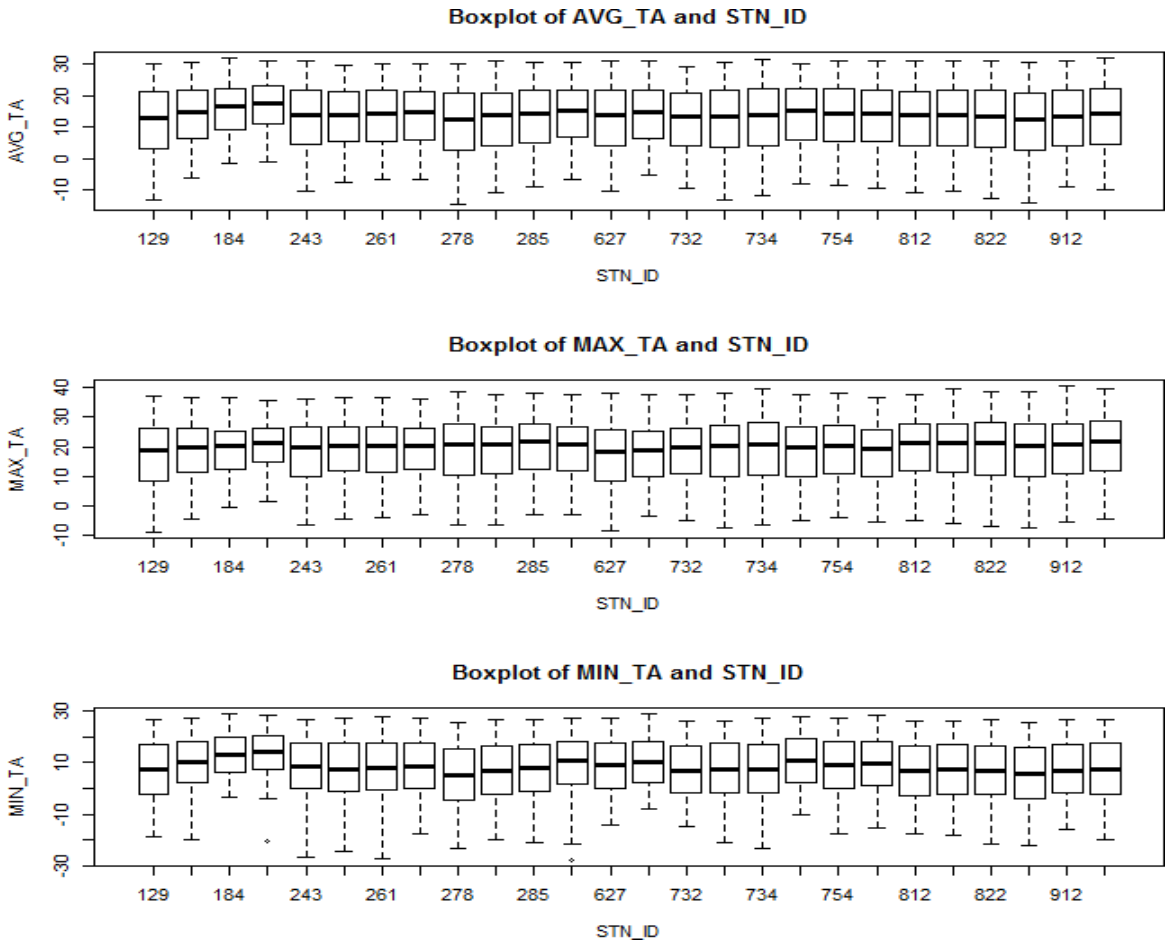
```
> head(subset(TA_RN_SS, is.na(AVG_TA) & is.na(MAX_TA) & is.na(MIN_TA)),3)
   STN_ID        TMA      YM AVG_TA MAX_TA MIN_TA SUM_RN SUM_SS_HR
1:    129 2000-01-26 2000-01     NA     NA     NA     NA       4.4
2:    129 2003-04-12 2003-04     NA     NA     NA     NA       0.0
3:    129 2012-06-01 2012-06     NA     NA     NA     NA       8.4
>
> summary(subset(TA_RN_SS, is.na(SUM_RN))); str(subset(TA_RN_SS, is.na(SUM_RN)))
     STN_ID       TMA                 YM          AVG_TA          MAX_TA
 732    : 6   Length:30          2001-08: 2   Min.   :-1.50   Min.   : 5.8
 129    : 5   Class :character   2001-10: 2   1st Qu.: 3.88   1st Qu.:11.0
 733    : 4   Mode  :character   2002-02: 2   Median :10.40   Median :14.2
 714    : 2                      2003-04: 2   Mean   :11.69   Mean   :16.9
 734    : 2                      2012-06: 2   3rd Qu.:19.57   3rd Qu.:24.1
 170    : 1                      2015-03: 2   Max.   :21.60   Max.   :27.4
 (Other):10                      (Other):18   NA's   :16      NA's   :15
     MIN_TA          SUM_RN      SUM_SS_HR
 Min.   :-5.20   Min.   : NA   Min.   : 0.00
 1st Qu.: 1.05   1st Qu.: NA   1st Qu.: 0.75
 Median : 8.00   Median : NA   Median : 5.95
 Mean   : 8.78   Mean   :NaN   Mean   : 5.43
 3rd Qu.:16.60   3rd Qu.: NA   3rd Qu.: 8.65
 Max.   :20.20   Max.   : NA   Max.   :12.40
 NA's   :15      NA's   :30    NA's   :18
Classes 'data.table' and 'data.frame':  30 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 1 1 1 1 1 2 3 4 5 7 ...
 $ TMA      : chr  "2000-01-26" "2003-04-12" "2009-04-25" "2009-06-11" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 1 40 112 114 150 89 150 162 184 174 .
 $ AVG_TA   : num  NA NA 9 19.5 NA 20.7 NA 21.6 7.7 19.6 ...
 $ MAX_TA   : num  NA NA 11.3 23.9 NA 26 NA 24.2 12 21.7 ...
 $ MIN_TA   : num  NA NA 7.9 13.8 NA 15.4 NA 19.8 3.8 17.8 ...
 $ SUM_RN   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SUM_SS_HR: num  4.4 0 0.3 12.4 8.4 12.1 9.4 0.9 7.5 0.3 ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> summary(subset(TA_RN_SS, is.na(SUM_SS_HR))); str(subset(TA_RN_SS, is.na(SUM_SS_HR)))
     STN_ID          TMA                  YM            AVG_TA
 627    : 5844   Length:82457       2000-08:  651   Min.   :-14.2
 714    : 5844   Class :character   2000-09:  630   1st Qu.:  4.5
 732    : 5844   Mode  :character   2000-10:  613   Median : 13.9
 733    : 5844                      2007-12:  451   Mean   : 13.1
 734    : 5844                      2000-01:  434   3rd Qu.: 21.7
 742    : 5844                      2000-03:  434   Max.   : 31.7
 (Other):47393                      (Other):79244   NA's   :307
     MAX_TA          MIN_TA           SUM_RN          SUM_SS_HR
 Min.   :-8.3   Min.   :-23.40   Min.   :  0.00   Min.   : NA
 1st Qu.:10.5   1st Qu.: -0.80   1st Qu.:  0.00   1st Qu.: NA
 Median :20.4   Median :  8.10   Median :  0.00   Median : NA
 Mean   :18.9   Mean   :  8.11   Mean   :  3.51   Mean   :NaN
 3rd Qu.:27.1   3rd Qu.: 17.50   3rd Qu.:  0.50   3rd Qu.: NA
 Max.   :40.4   Max.   : 28.80   Max.   :384.50   Max.   : NA
 NA's   :18     NA's   :13       NA's   :18       NA's   :82457
```

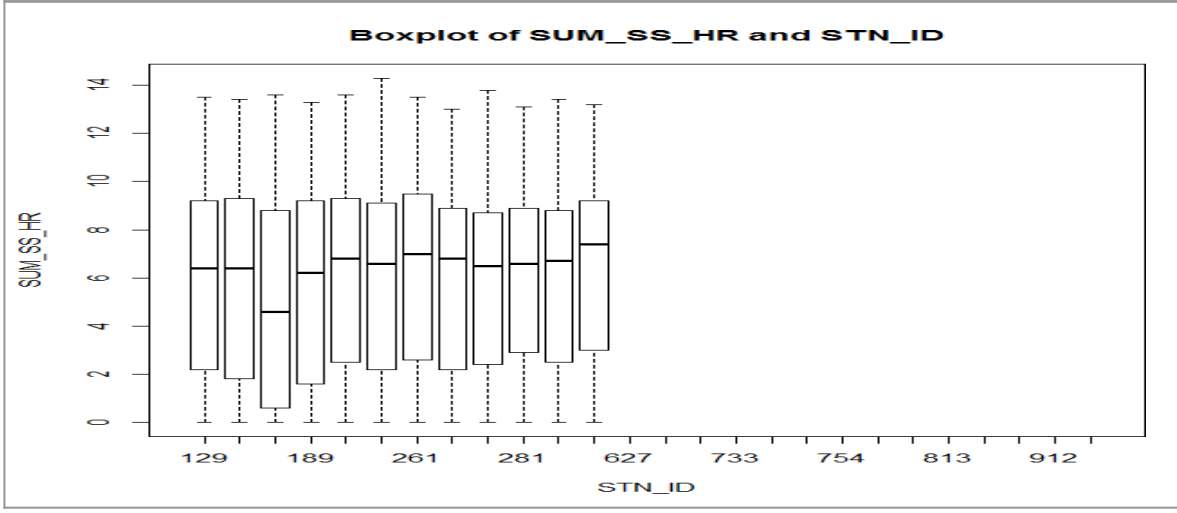
2. 탐색적 자료 분석(6/9)

```
Classes 'data.table' and 'data.frame':  82457 obs. of  8 variables:
 $ STN_ID   : Factor w/ 26 levels "129","170","184",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ TMA      : chr  "2007-12-04" "2007-12-05" "2007-12-06" "2007-12-07" ...
 $ YM       : Factor w/ 192 levels "2000-01","2000-02",..: 96 96 96 96 96 96 96 96 96 96 ...
 $ AVG_TA   : num  7.2 7.2 7.8 9.4 8.8 8.1 10.8 13.1 10.1 9 ...
 $ MAX_TA   : num  9.3 8.4 9.8 10.8 10.3 12 14.6 15.3 10.7 10.5 ...
 $ MIN_TA   : num  6.4 6.3 6.3 7.5 7.1 5.5 5.5 10.7 9.2 7.5 ...
 $ SUM_RN   : num  0 0 0.5 0.5 0 0 6 0 6.5 0 ...
 $ SUM_SS_HR: num  NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, ".internal.selfref")=<externalptr>
>
> # Boxplot
> par(mfrow=c(3,1))
> boxplot(data=TA_RN_SS, AVG_TA~STN_ID, xlab="STN_ID", ylab="AVG_TA", main="Boxplot of AVG_TA an
> boxplot(data=TA_RN_SS, MAX_TA~STN_ID, xlab="STN_ID", ylab="MAX_TA", main="Boxplot of MAX_TA an
> boxplot(data=TA_RN_SS, MIN_TA~STN_ID, xlab="STN_ID", ylab="MIN_TA", main="Boxplot of MIN_TA an
>
> par(mfrow=c(1,1))
> boxplot(data=TA_RN_SS, SUM_SS_HR~STN_ID, xlab="STN_ID", ylab="SUM_SS_HR", main="Boxplot of SUM
>
> ## onion
> summary(onion)
 region_1       year            y
 경남:15   Min.   :2001   Min.   :3936
 경북:15   1st Qu.:2004   1st Qu.:5600
 전남:15   Median :2008   Median :6186
 전북:15   Mean   :2008   Mean   :6138
 제주:15   3rd Qu.:2012   3rd Qu.:6611
 충남:15   Max.   :2015   Max.   :8386
>
> boxplot(data=onion, y~region_1, xlab="region_1", ylab="y", main="Boxplot of y and region_1" )
> boxplot(data=onion, y~year, xlab="year", ylab="y", main="Boxplot of y and year" )
>
> ## onion.area
> summary(onion.area)
     STN_ID    region_1    region_2        area_id        area
 129    : 1   경남:4   고흥   : 1   4400000000:2   Min.   :  83
 170    : 1   경북:6   고흥   : 1   4500000000:3   1st Qu.: 325
 184    : 1   전남:9   군위   : 1   4600000000:9   Median : 523
 189    : 1   전북:3   김천   : 1   4700000000:6   Mean   : 760
 243    : 1   제주:2   남해   : 1   4800000000:4   3rd Qu.:1040
 260    : 1   충남:2   무안   : 1   5000000000:2   Max.   :3355
 (Other):20            (Other):20                  NA's   :8
       w
 Min.   :0.000
 1st Qu.:0.054
 Median :0.174
 Mean   :0.231
 3rd Qu.:0.404
 Max.   :0.577

>
> boxplot(data=onion.area, area~region_1, xlab="region_1", ylab="area", main="Boxplot of area an
>
> qplot(x=w, y=area, data=subset(onion.area, !is.na(w) & !is.na(area)), color=region_1, geom = c
> #============================================================#
```

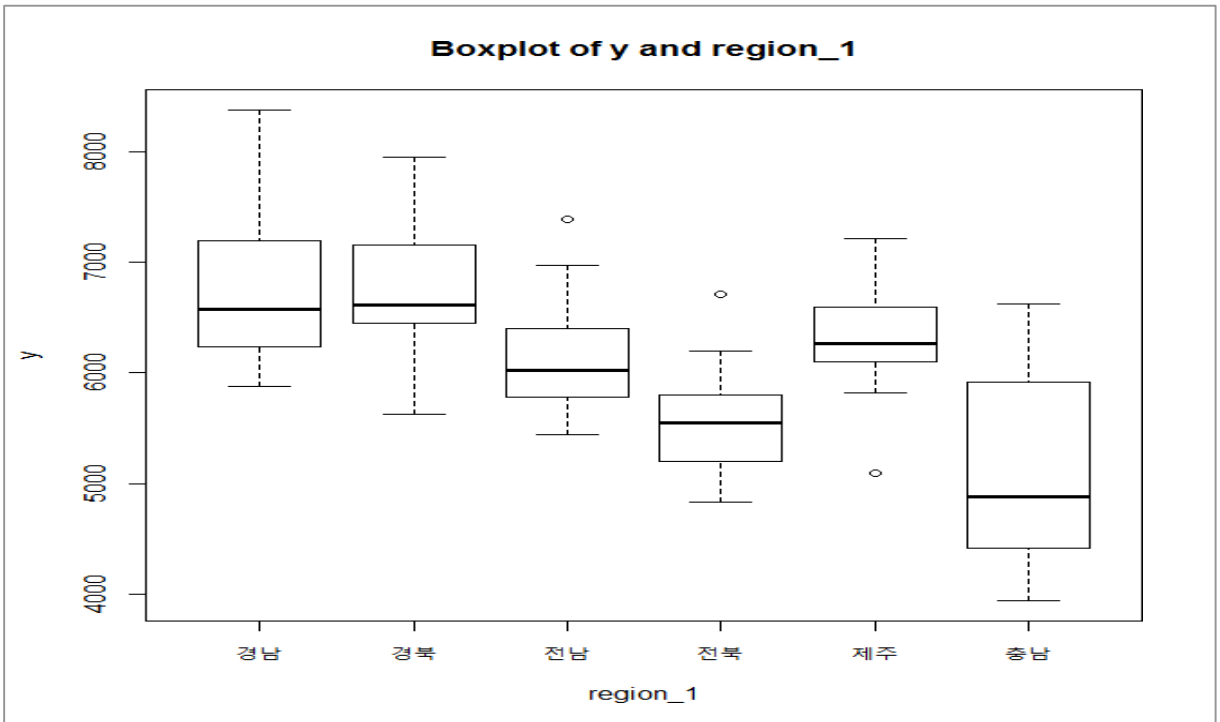
2. 탐색적 자료 분석(7/9)

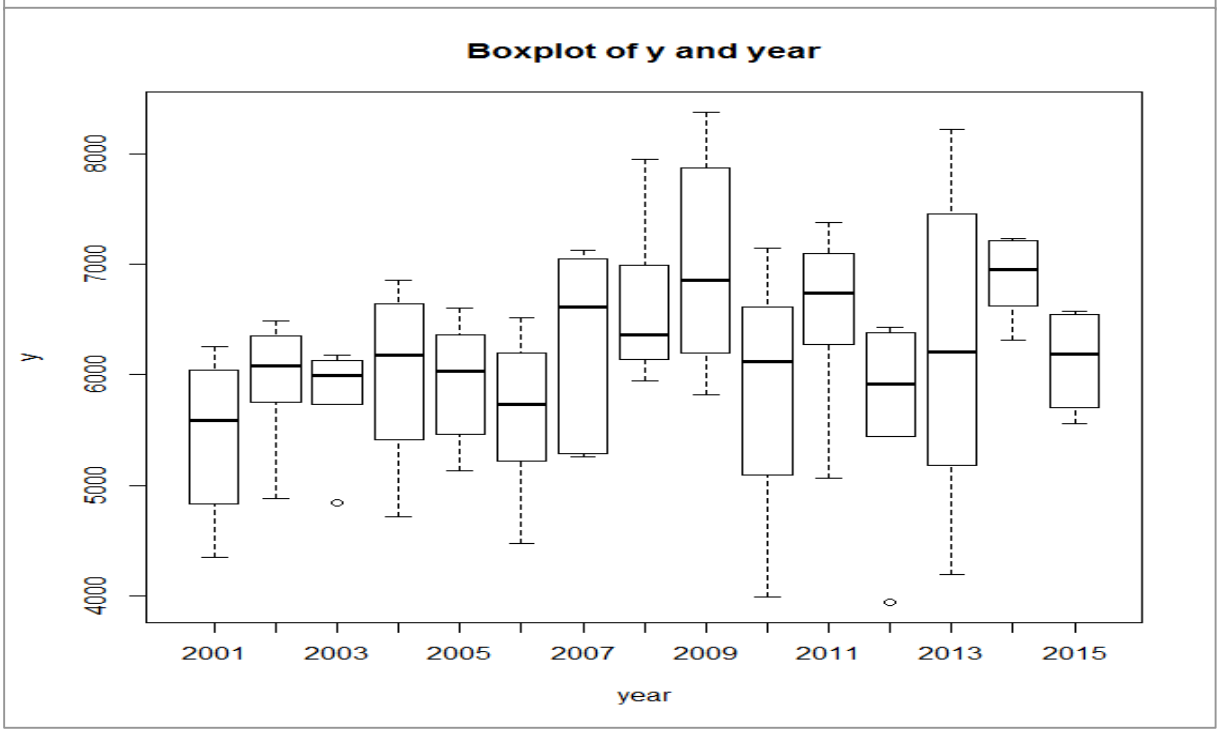
> boxplot 실행 결과



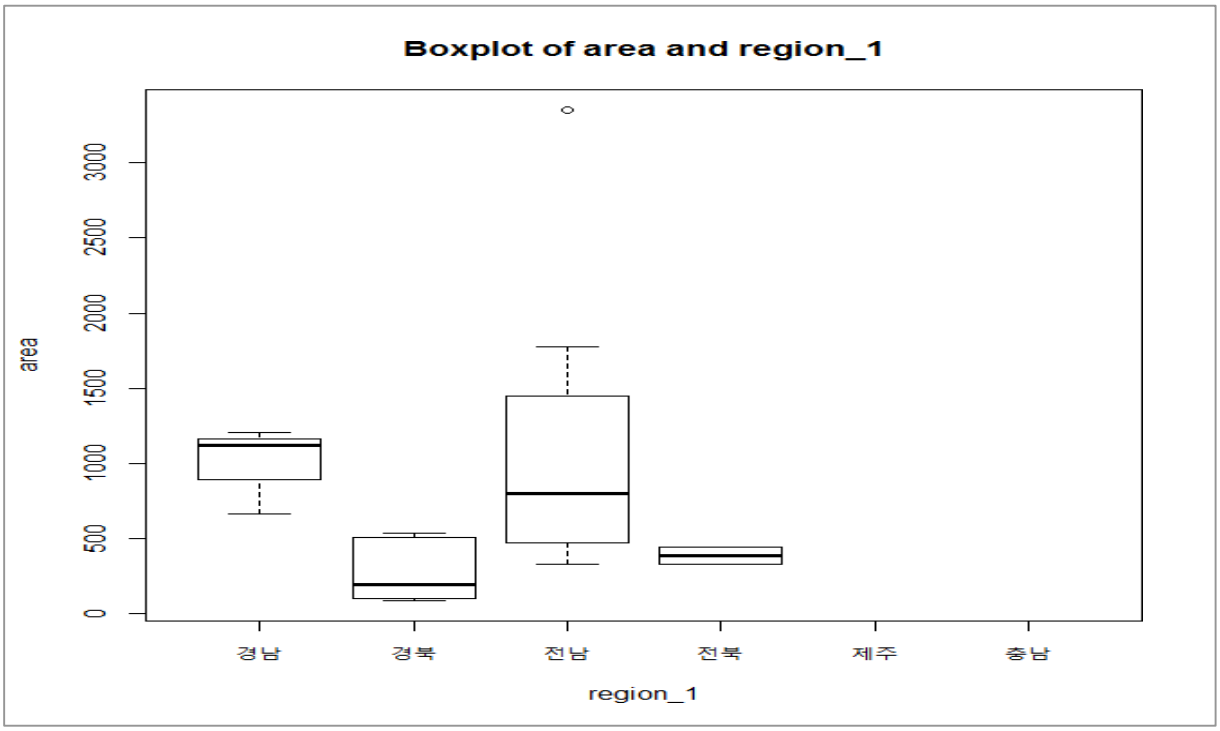


2. 탐색적 자료 분석(8/9)

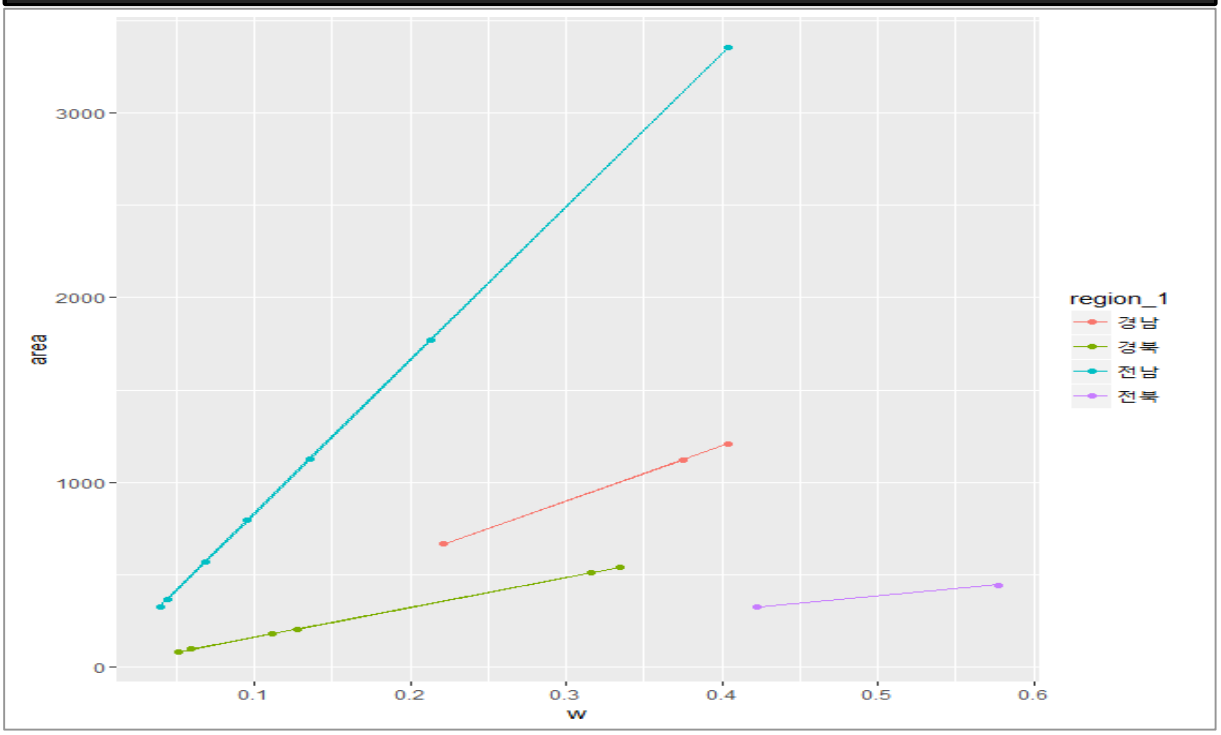

Boxplot of y and region_1
y
4000
5000
6000
7000
8000
경남
경북
전남
전북
제주
충남
region_1


Boxplot of y and year
y
4000
5000
6000
7000
8000
2001
2003
2005
2007
2009
2011
2013
2015
year

2. 탐색적 자료 분석(9/9)



> qplot 실행 결과



기상청

Ⅲ. 데이터처리

1. 이상치 및 결측치 대체
2. 파생변수 생성
3. 분석 데이터셋 완성

1. 이상치 및 결측치 대체 (1/4)

- **이상치 정의**

- 기상기후 품질 알고리즘 기준값

항목	요소	물리한계검사 상한	물리한계검사 하한
AWS	기온(℃)	45	-35
	일강수량(㎜)	1500	0
ASOS	가조시간(hr)	15	0

- 기상기후 품질 알고리즘 기준값에 의거 한 기상 데이터 이상치 (outlier) 처리

```
#=============================================================#
#Define Outlier
#=============================================================#
##TA_RN_SS
#Using Algorithm
TA_RN_SS[AVG_TA <-35| AVG_TA >45, AVG_TA := NA]
TA_RN_SS[MAX_TA <-35| MAX_TA >45, MAX_TA :=NA]
TA_RN_SS[MIN_TA <-35|MIN_TA >45, MIN_TA :=NA]

TA_RN_SS[SUM_RN <0 |SUM_RN >1500, SUM_RN := NA]

TA_RN_SS[SUM_SS_HR <0 |SUM_SS_HR >15, SUM_SS_HR := NA]
#=============================================================#
```

- **이상치 및 결측치 대체**

- 정의 된 이상치 및 EDA 결과 비논리적 또는 이상치 (outlier)로 보이는 값대체

```
#=============================================================#
#Replace Outlier
#=============================================================#
##onion
onion[, region_1 :=as.character(region_1)]

#Using Boxplot
onion[, y := as.numeric(y)]
onion$y.replace <-onion$y
```

1. 이상치 및 결측치 대체 (2/4)

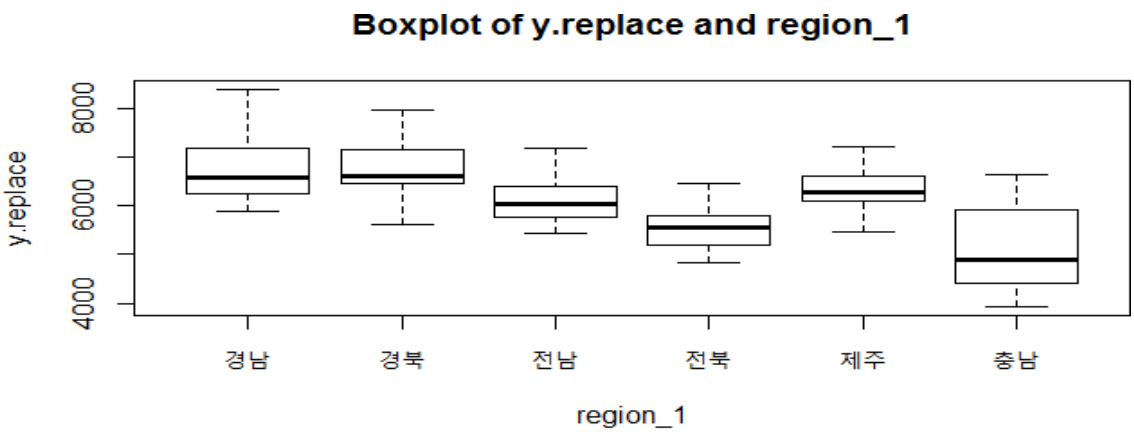
```
tmp <- onion[, .SD[which.max(y)], by = region_1, .SDcols="y"]
onion[region_1==“전남" & y==as.numeric(subset(tmp, region_1 ==“전남", select=y)),
y.replace := NA]
tmp <- onion[, .SD[which.max(y.replace)], by = region_1, .Sdcols="y.replace"]
onion[region_1=="전남" & is.na(y.replace), y.replace :=(y+as.numeric(subset(tmp,
region_1=="전남", select=y.replace)) )/2 ]

tmp <- onion[, .SD[which.max(y)], by = region_1, .SDcols="y"]
onion[region_1=="전북" & y==as.numeric(subset(tmp, region_1=="전북", select=y)),
y.replace := NA]
tmp <- onion[, .SD[which.max(y.replace)], by = region_1, .SDcols="y.replace"]
onion[region_1=="전북" & is.na(y.replace), y.replace :=(y+as.numeric(subset(tmp,
region_1=="전북", select=y.replace)) )/2 ]

tmp <- onion[, .SD[which.min(y)], by = region_1, .SDcols="y"]
onion[region_1=="제주" & y==as.numeric(subset(tmp, region_1=="제주", select=y)),
y.replace := NA]
tmp <- onion[, .SD[which.min(y.replace)], by = region_1, .SDcols="y.replace"]
onion[region_1=="제주" & is.na(y.replace), y.replace :=(y+as.numeric(subset(tmp,
region_1=="제주", select=y.replace)) )/2 ]

par(mfrow=c(2,1))
boxplot(data=onion, y.replace~region_1, xlab="region_1", ylab="y.replace", main="Boxplot of
y.replace and region_1" )
boxplot(data=onion, y~region_1, xlab="region_1", ylab="y", main="Boxplot of y and
region_1" )

##onion.area
onion.area[, STN_ID := as.character(STN_ID)]
onion.area[, region_1 := as.character(region_1)]
onion.area[, region_2 := as.character(region_2)]
onion.area[, area_id := as.character(area_id)]
onion.area[, area := as.numeric(area)]
onion.area[, area.replace := as.numeric(area)]
str(onion.area)

tmp <- onion.area[, .SD[which.max(area)], by = region_1, .SDcols="area"]
onion.area[is.na(area.replace), area.replace := 0]
onion.area[region_1== " 전남" & area==as.numeric(subset(tmp, region_1== " 전남",
select=area)), area.replace := NA]
tmp <-onion.area[, .SD[which.max(area.replace)], by = region_1, .SDcols="area.replace"]
onion.area[region_1== " 전남" & is.na(area.replace), area.replace :=
(area+as.numeric(subset(tmp, region_1== " 전남", select=area.replace)) )/2 ]
onion.area[area.replace==0, area.replace := NA]
```

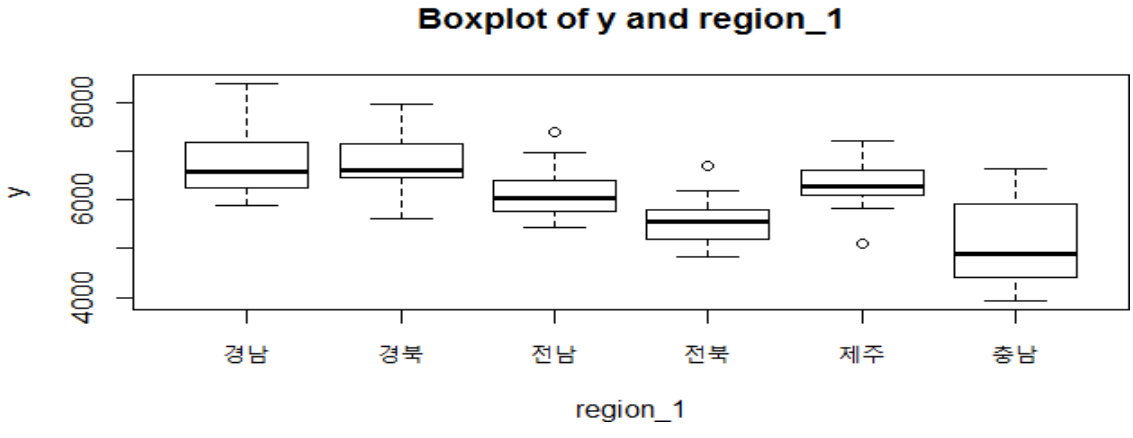
1. 이상치 및 결측치 대체 (3/4)

```
par(mfrow=c(2,1))
boxplot(data=onion.area
boxplot(data=onion.area, area.replace~region_1, xlab="region_1", ylab="area.replace",
main="Boxplot of area.replace and region_1")
boxplot(data=onion.area, area~region_1, xlab="region_1", ylab="area", main="Boxplot of area
and region_1")
rm(tmp)
#=================================================================#
```

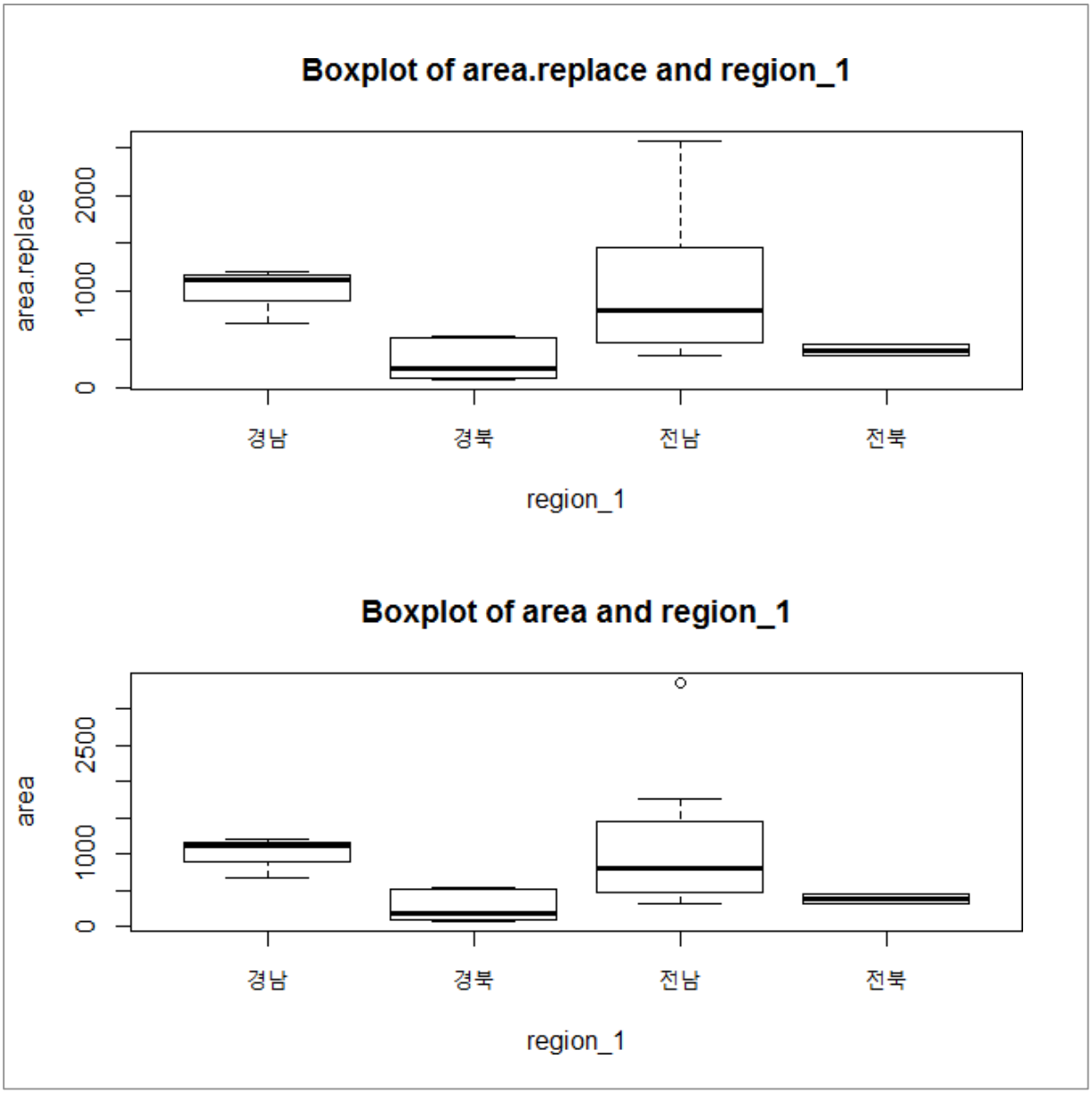
- which.max() 로 숫자형 변수에서 가장 큰 값만표시
- which.min() 로 숫자형 변수에서 가장 작은 값만표시
- NA 로 결측치 입력

> boxplot 실행 결과





1. 이상치 및 결측치 대체 (4/4)



2. 파생변수 생성(1/3)

- **파생변수 생성**

- 기온 일교차 및 기상요소의 월 평균값, 최대값, 최소값, 합계값 생성

```
#=============================================================#
#Creating Derivated Variables
#=============================================================#
##TA_RN_SS
TA_RN_SS[, STN_ID := as.character(STN_ID)]
TA_RN_SS[, YM  :=as.character(YM)]

#Creating Derivated Variables of DTD
TA_RN_SS[, DTD := MAX_TA  - MIN_TA]

# mean, max, min, sum
TA_RN_SS_Month <- ddply(TA_RN_SS, c("STN_ID", "YM"), summarise,
            TAD=mean(AVG_TA, na.rm=TRUE), TAmax=max(MAX_TA, na.rm=TRUE),
            TAmin=min(MIN_TA, na.rm=TRUE), RAINSUM=sum(SUM_RN, na.rm=TRUE),
            DTD=mean(DTD, na.rm=TRUE), SUM_SS_HR=mean(SUM_SS_HR,  na.rm=TRUE))
str(TA_RN_SS_Month)
summary(TA_RN_SS_Month)
rm(TA_RN_SS)

# Replace Outlier of SUM_SS_HR
TA_RN_SS_Month <- as.data.table(TA_RN_SS_Month)
TA_RN_SS_Month[, SUM_SS_HR.replace := mean(SUM_SS_HR, na.rm=TRUE), by=c("YM")]
TA_RN_SS_Month[!is.na(SUM_SS_HR), SUM_SS_HR.replace := SUM_SS_HR]
TA_RN_SS_Month[, SUM_SS_HR := NULL]
setnames(TA_RN_SS_Month, "SUM_SS_HR.replace", "SUM_SS_HR")
str(TA_RN_SS_Month)
summary(TA_RN_SS_Month)
#=============================================================#
```

- as.character()로 STN_ID, YM 변수를 문자열 타입으로 변환
- ddply()로 STN_ID, YM 을 기준으로 컬럼을 나누어 mean, min, max, sum 함수를 적용함
- mean()으로 월 평균 기온, 월 평균 일교차, 월 평균 일조시간을계산
- min()으로 월 최소 기온을계산
- max()로 월 최대 기온을계산
- sum()으로 월 평균 누적강수량을계산
- 결측이 있는 일조시간 변수를 전체 해당 월 평균 일조시간으로 대체

2. 파생변수 생성(2/3)

> 실행 결과

```
> #==================================================================#
> # Creating Derivated Variables
> #==================================================================#
> ## TA_RN_SS
> TA_RN_SS[, STN_ID := as.character(STN_ID)]
> TA_RN_SS[, YM := as.character(YM)]
>
> # Creating Derivated Variables of DTD
> TA_RN_SS[, DTD := MAX_TA - MIN_TA]
>
> # mean, max, min, sum
> TA_RN_SS_Month <- ddply(TA_RN_SS, c("STN_ID", "YM"), summarise,
+              TAD=mean(AVG_TA, na.rm=TRUE), TAmax=max(MAX_TA, na.rm=TRUE),
+              TAmin=min(MIN_TA, na.rm=TRUE), RAINSUM=sum(SUM_RN, na.rm=TRUE),
+              DTD=mean(DTD, na.rm=TRUE), SUM_SS_HR=mean(SUM_SS_HR, na.rm=TRUE))
> str(TA_RN_SS_Month)
'data.frame':   4992 obs. of  8 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.552 -2.572 3.857 9.49 15.019 ...
 $ TAmax    : num  21.4 7.4 17.2 21.4 29.6 30.9 32.2 32.7 29.7 26.5 ...
 $ TAmin    : num  -13.3 -12.9 -9.8 -4 4.5 11.7 12.9 17.7 9.2 -0.3 ...
 $ RAINSUM  : num  58 0.5 2 36.5 71.5 215 61.5 647 323 33.2 ...
 $ DTD      : num  8.11 10.43 13.66 12.56 10.23 ...
 $ SUM_SS_HR: num  4.08 6.66 7.35 7.72 6.7 ...
> summary(TA_RN_SS_Month)
    STN_ID               YM                 TAD              TAmax
 Length:4992        Length:4992        Min.   :-7.02   Min.   : 3.4
 Class :character   Class :character   1st Qu.: 5.41   1st Qu.:19.0
 Mode  :character   Mode  :character   Median :13.72   Median :26.7
                                       Mean   :13.33   Mean   :25.3
                                       3rd Qu.:21.69   3rd Qu.:32.0
                                       Max.   :29.14   Max.   :40.4

     TAmin            RAINSUM            DTD          SUM_SS_HR
 Min.   :-27.80   Min.   :   0.0   Min.   : 3.38   Min.   : 1.03
 1st Qu.: -6.33   1st Qu.:  29.5   1st Qu.: 8.20   1st Qu.: 4.82
 Median :  1.30   Median :  66.0   Median :10.23   Median : 5.84
 Mean   :  2.53   Mean   : 112.8   Mean   :10.44   Mean   : 5.85
 3rd Qu.: 12.00   3rd Qu.: 147.0   3rd Qu.:12.57   3rd Qu.: 6.86
 Max.   : 24.60   Max.   :1152.5   Max.   :22.01   Max.   :11.89
                                                   NA's   :2702
> rm(TA_RN_SS)
```

2. 파생변수 생성(3/3)

```
> # Replace Outlier of SUM_SS_HR
> TA_RN_SS_Month <- as.data.table(TA_RN_SS_Month)
> TA_RN_SS_Month[, SUM_SS_HR.replace := mean(SUM_SS_HR, na.rm=TRUE), by=c("YM")]
> TA_RN_SS_Month[!is.na(SUM_SS_HR), SUM_SS_HR.replace := SUM_SS_HR]
> TA_RN_SS_Month[, SUM_SS_HR := NULL]
> setnames(TA_RN_SS_Month, "SUM_SS_HR.replace", "SUM_SS_HR")
> str(TA_RN_SS_Month)
Classes 'data.table' and 'data.frame':  4992 obs. of  8 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.552 -2.572 3.857 9.49 15.019 ...
 $ TAmax    : num  21.4 7.4 17.2 21.4 29.6 30.9 32.2 32.7 29.7 26.5 ...
 $ TAmin    : num  -13.3 -12.9 -9.8 -4 4.5 11.7 12.9 17.7 9.2 -0.3 ...
 $ RAINSUM  : num  58 0.5 2 36.5 71.5 215 61.5 647 323 33.2 ...
 $ DTD      : num  8.11 10.43 13.66 12.56 10.23 ...
 $ SUM_SS_HR: num  4.08 6.66 7.35 7.72 6.7 ...
 - attr(*, ".internal.selfref")=<externalptr>
> summary(TA_RN_SS_Month)
    STN_ID               YM                  TAD             TAmax
 Length:4992        Length:4992        Min.   :-7.02   Min.   : 3.4
 Class :character   Class :character   1st Qu.: 5.41   1st Qu.:19.0
 Mode  :character   Mode  :character   Median :13.72   Median :26.7
                                       Mean   :13.33   Mean   :25.3
                                       3rd Qu.:21.69   3rd Qu.:32.0
                                       Max.   :29.14   Max.   :40.4
     TAmin           RAINSUM            DTD          SUM_SS_HR
 Min.   :-27.80   Min.   :   0.0   Min.   : 3.38   Min.   : 1.03
 1st Qu.: -6.33   1st Qu.:  29.5   1st Qu.: 8.20   1st Qu.: 4.90
 Median :  1.30   Median :  66.0   Median :10.23   Median : 5.82
 Mean   :  2.53   Mean   : 112.8   Mean   :10.44   Mean   : 5.85
 3rd Qu.: 12.00   3rd Qu.: 147.0   3rd Qu.:12.57   3rd Qu.: 6.82
 Max.   : 24.60   Max.   :1152.5   Max.   :22.01   Max.   :11.89
```

3. 최종 데이터셋 완성(1/8)

- **기상 데이터 가중치 적용 및 지역별 기상변수생성**

- 양파 재배면적 데이터와 병합하여 재배 면적비율 가중치를 적용

- 지역별 기상변수 생성

```
#============================================================#
#Creating Dataset of Analysis
#============================================================#
#TA_RN_SS_Month and onion.area
Weather_Area <-merge(TA_RN_SS_Month, onion.area, by="STN_ID", all.x=TRUE, all.y=FALSE)
Weather_Area <-subset(Weather_Area, !is.na(w) & w >0)
Weather_Area[, area := NULL]
setnames(Weather_Area, "area.replace", "area")
str(Weather_Area)

#from STN_ID to region_1
Weather_Area[, TAD:=as.numeric(TAD)*w]
Weather_Area[, TAmax:=as.numeric(TAmax)*w]
Weather_Area[, TAmin:=as.numeric(TAmin)*w]
Weather_Area[, DTD:=as.numeric(DTD)*w]
Weather_Area[, RAINSUM:=as.numeric(RAINSUM)*w]
Weather_Area[, SUM_SS_HR:=as.numeric(SUM_SS_HR)*w]
str(Weather_Area)

Weather_Area_Region <-ddply(Weather_Area, c("area_id", "region_1", "region_2", "YM"),
summarise,
        TAD=sum(TAD, na.rm=TRUE), TAmax=sum(TAmax, na.rm=TRUE),
        TAmin=sum(TAmin, na.rm=TRUE), DTD=sum(DTD, na.rm=TRUE),
        RAINSUM=sum(RAINSUM, na.rm=TRUE), SUM_SS_HR=sum(SUM_SS_HR,
na.rm=TRUE), area=sum(area, na.rm=TRUE))
str(Weather_Area_Region)
rm(Weather_Area, TA_RN_SS_Month, onion.area)
```

- **merge()로 파생변수가 포함된 기상 데이터와 양파 재배면적 데이터를 병합**
- **subset()으로 면적비율이 결측이거나 0 인 데이터제외**
- **기상 데이터에 면적비율을 곱하여 면적비율 가중치를 적용**
- **ddply()로 월별 및 지역별 평균 기온, 최소 기온, 최대 기온, 일교차, 누적 강수량, 합계 일조 시간, 합계 재배면적을 계산**

3. 최종 데이터셋 완성(2/8)

> 실행 결과

```
> #===============================================================#
> # Creating Dataset of Analysis
> #===============================================================#
> # TA_RN_SS_Month and onion.area
> Weather_Area <- merge(TA_RN_SS_Month, onion.area, by="STN_ID", all.x=TRUE, all.y=FALSE)
> Weather_Area <- subset(Weather_Area, !is.na(w) & w > 0 )
> Weather_Area[, area := NULL]
> setnames(Weather_Area, "area.replace", "area")
> str(Weather_Area)
Classes 'data.table' and 'data.frame':  4224 obs. of  13 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.552 -2.572 3.857 9.49 15.019 ...
 $ TAmax    : num  21.4 7.4 17.2 21.4 29.6 30.9 32.2 32.7 29.7 26.5 ...
 $ TAmin    : num  -13.3 -12.9 -9.8 -4 4.5 11.7 12.9 17.7 9.2 -0.3 ...
 $ RAINSUM  : num  58 0.5 2 36.5 71.5 215 61.5 647 323 33.2 ...
 $ DTD      : num  8.11 10.43 13.66 12.56 10.23 ...
 $ SUM_SS_HR: num  4.08 6.66 7.35 7.72 6.7 ...
 $ region_1 : chr  "충남" "충남" "충남" "충남" ...
 $ region_2 : chr  "서산" "서산" "서산" "서산" ...
 $ area_id  : chr  "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ w        : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ area     : num  NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, "sorted")= chr "STN_ID"
 - attr(*, ".internal.selfref")=<externalptr>
>
> # from STN_ID to region_1
> Weather_Area[, TAD:=as.numeric(TAD)*w]
> Weather_Area[, TAmax:=as.numeric(TAmax)*w]
> Weather_Area[, TAmin:=as.numeric(TAmin)*w]
> Weather_Area[, DTD:=as.numeric(DTD)*w]
> Weather_Area[, RAINSUM:=as.numeric(RAINSUM)*w]
> Weather_Area[, SUM_SS_HR:=as.numeric(SUM_SS_HR)*w]
> str(Weather_Area)
Classes 'data.table' and 'data.frame':  4224 obs. of  13 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.552 -2.572 3.857 9.49 15.019 ...
 $ TAmax    : num  21.4 7.4 17.2 21.4 29.6 30.9 32.2 32.7 29.7 26.5 ...
 $ TAmin    : num  -13.3 -12.9 -9.8 -4 4.5 11.7 12.9 17.7 9.2 -0.3 ...
 $ RAINSUM  : num  58 0.5 2 36.5 71.5 215 61.5 647 323 33.2 ...
 $ DTD      : num  8.11 10.43 13.66 12.56 10.23 ...
 $ SUM_SS_HR: num  4.08 6.66 7.35 7.72 6.7 ...
 $ region_1 : chr  "충남" "충남" "충남" "충남" ...
 $ region_2 : chr  "서산" "서산" "서산" "서산" ...
 $ area_id  : chr  "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ w        : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ area     : num  NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, "sorted")= chr "STN_ID"
 - attr(*, ".internal.selfref")=<externalptr>
```

3. 최종 데이터셋 완성(3/8)

```
> # from STN_ID to region_1
> Weather_Area[, TAD:=as.numeric(TAD)*w]
> Weather_Area[, TAmax:=as.numeric(TAmax)*w]
> Weather_Area[, TAmin:=as.numeric(TAmin)*w]
> Weather_Area[, DTD:=as.numeric(DTD)*w]
> Weather_Area[, RAINSUM:=as.numeric(RAINSUM)*w]
> Weather_Area[, SUM_SS_HR:=as.numeric(SUM_SS_HR)*w]
> str(Weather_Area)
Classes 'data.table' and 'data.frame':  4224 obs. of  13 variables:
 $ STN_ID   : chr  "129" "129" "129" "129" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.276 -1.286 1.928 4.745 7.51 ...
 $ TAmax    : num  10.7 3.7 8.6 10.7 14.8 ...
 $ TAmin    : num  -6.65 -6.45 -4.9 -2 2.25 5.85 6.45 8.85 4.6 -0.15 ...
 $ RAINSUM  : num  29 0.25 1 18.25 35.75 ...
 $ DTD      : num  4.05 5.21 6.83 6.28 5.11 ...
 $ SUM_SS_HR: num  2.04 3.33 3.67 3.86 3.35 ...
 $ region_1 : chr  "충남" "충남" "충남" "충남" ...
 $ region_2 : chr  "서산" "서산" "서산" "서산" ...
 $ area_id  : chr  "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ w        : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ area     : num  NA NA NA NA NA NA NA NA NA NA ...
 - attr(*, "sorted")= chr "STN_ID"
 - attr(*, ".internal.selfref")=<externalptr>
> Weather_Area_Region <- ddply(Weather_Area, c("area_id", "region_1", "region_2", "YM"), summar
+             TAD=sum(TAD, na.rm=TRUE), TAmax=sum(TAmax, na.rm=TRUE),
+             TAmin=sum(TAmin, na.rm=TRUE), DTD=sum(DTD, na.rm=TRUE),
+             RAINSUM=sum(RAINSUM, na.rm=TRUE), SUM_SS_HR=sum(SUM_SS_HR, na.rm=TRUE),
+             area=sum(area, na.rm=TRUE))
str(Weather_Area_Region)
> 'data.frame': 4224 obs. of  11 variables:
 $ area_id  : chr  "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ region_1 : chr  "충남" "충남" "충남" "충남" ...
 $ region_2 : chr  "서산" "서산" "서산" "서산" ...
 $ YM       : chr  "2000-01" "2000-02" "2000-03" "2000-04" ...
 $ TAD      : num  -0.276 -1.286 1.928 4.745 7.51 ...
 $ TAmax    : num  10.7 3.7 8.6 10.7 14.8 ...
 $ TAmin    : num  -6.65 -6.45 -4.9 -2 2.25 5.85 6.45 8.85 4.6 -0.15 ...
 $ DTD      : num  4.05 5.21 6.83 6.28 5.11 ...
 $ RAINSUM  : num  29 0.25 1 18.25 35.75 ...
 $ SUM_SS_HR: num  2.04 3.33 3.67 3.86 3.35 ...
 $ area     : num  0 0 0 0 0 0 0 0 0 0 ...
> rm(Weather_Area, TA_RN_SS_Month, onion.area)
```

3. 최종 데이터셋 완성(4/8)

- **월 변수 생성 및 양파 생육시기를 고려한 기상 데이터처리**

- 양파는 8월에 재배하기 때문에 양파의 생육시기를 고려한 기상 데이터 처리가 필요
- 이번 년도의 양파 생산량을 예측하기 위해서 전년도 9, 10, 11, 12월의 기상 상태를 이번 년도로 적용하여 분석하도록 데이터 조정

```
#Separating Month
Weather_Area_Region <-melt(Weather_Area_Region, id.var=c("area_id", "region_1",
"region_2", "YM", "area"))
Weather_Area_Region <-as.data.table(Weather_Area_Region)
Weather_Area_Region[, year := substring(YM,1,4)]
Weather_Area_Region[, month := substring(YM,6,7)]
Weather_Area_Region <-cast(Weather_Area_Region,
area_id+region_1+region_2+year+area~variable+month)
str(Weather_Area_Region)

#ShiftMonth
Weather_Area_Region <-data.table(Weather_Area_Region)
Weather_Area_Region[, year := as.numeric(year)]
str(Weather_Area_Region)

Weather_Area_Region <-as.data.frame(Weather_Area_Region)
Weather_Area_Region <-Weather_Area_Region[do.call("order",
Weather_Area_Region[c("region_1", "region_2", "year")]),]
index <-word(names(Weather_Area_Region),-1, sep="_") %in% c('09','10','11','12')
Weather_Area_Region[2:nrow(Weather_Area_Region),index] <-
Weather_Area_Region[1:(nrow(Weather_Area_Region)-1),index]
Weather_Area_Region <-as.data.table(Weather_Area_Region)
rm(index)

tmp <-Weather_Area_Region[, .SD[which.min(year)], by=c("region_1",
"region_2"), .SDcols="year"]
tmp[, remove := 1]

Weather_Area_Region <- as.data.frame(Weather_Area_Region)
tmp <- as.data.frame(tmp)
Weather_Area_Region <-merge(Weather_Area_Region, tmp, by=c("region_1", "region_2",
"year"), all.x=TRUE, all.y=FALSE)
rm(tmp)
Weather_Area_Region <-subset(Weather_Area_Region, is.na(remove))
Weather_Area_Region <-as.data.table(Weather_Area_Region)
Weather_Area_Region[, remove := NULL]
str(Weather_Area_Region)
head(Weather_Area_Region, 5)
```

3. 최종 데이터셋 완성(5/8)

```
#Weather_Area_Region and onion
Weather_Area_Region[, year := as.integer(year)]

Weather_Area_Region <-as.data.frame(Weather_Area_Region)
onion <-as.data.frame(onion)
Dataset <-merge(Weather_Area_Region, onion, by=c("region_1", "year"), all.x=TRUE,
all.y=FALSE)
rm(Weather_Area_Region, onion)

Dataset <-as.data.table(Dataset)
Dataset <-subset(Dataset, !is.na(y.replace))
Dataset[, y := NULL]
setnames(Dataset, "y.replace", "y" )
str(Dataset)

#Re-order columns
names(Dataset)
Dataset <-subset(Dataset, select=c(area_id, region_1, region_2, area, year, y,
TAD_01:SUM_SS_HR_12))
str(Dataset)

#Check
ls()
tables()
gc()
```

- **melt()로 기상변수를 하나의 variable 컬럼으로 통합(행으로 표현)**
- **year과 month 컬럼 생성**
- **cast()로 melt된 데이터를 다시 variable+month를 기준으로 컬럼으로 변환**
- **order()로 region_1, region_2, year 별로 데이터를 정렬**
- **word()로 9,10,11,12월의 컬럼을 선택**
- **9, 10, 11, 12월에 해당되는 기상변수를 1년씩 뒤로 미뤄 다음 년도 기상 데이터와 매핑**
- **데이터 중 가장 최소 연도의 데이터를 삭제**
- **merge()로 양파 생산량 데이터와 병합**
- **subset()으로 양파 생산량이 있는 데이터만 추출**
- **subset()으로 분석에 필요한 컬럼만 추출하여 최종 데이터셋 완성**

3. 최종 데이터셋 완성(6/8)

> 실행 결과

```
> # Separating Month
> Weather_Area_Region <- melt(Weather_Area_Region, id.var=c("area_id", "region_1", "region_2", "
> Weather_Area_Region <- as.data.table(Weather_Area_Region)
> Weather_Area_Region[, year := substring(YM,1,4)]
> Weather_Area_Region[, month := substring(YM,6,7)]
> Weather_Area_Region <- cast(Weather_Area_Region, area_id+region_1+region_2+year+area~variable+
> str(Weather_Area_Region)
List of 77
 $ area_id     : chr [1:352] "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ region_1    : chr [1:352] "충남" "충남" "충남" "충남" ...
 $ region_2    : chr [1:352] "서산" "서산" "서산" "서산" ...
 $ year        : chr [1:352] "2000" "2001" "2002" "2003" ...
 $ area        : num [1:352] 0 0 0 0 0 0 0 0 0 0 ...
 $ TAD_01      : num [1:352] -0.276 -1.948 0.145 -1.768 -0.934 ...
 $ TAD_02      : num [1:352] -1.286 -0.421 0.157 0.448 0.81 ...
 $ TAD_03      : num [1:352] 1.93 1.69 2.92 2.28 2.43 ...
 $ TAD_04      : num [1:352] 4.74 5.51 6.16 5.93 5.53 ...
 $ TAD_05      : num [1:352] 7.51 8.74 8.14 9.11 8.3 ...
 $ TAD_06      : num [1:352] 10.4 10.7 10.5 10.4 11 ...
 $ TAD_07      : num [1:352] 12.5 12.7 12.3 11.8 12.4 ...
 $ TAD_08      : num [1:352] 12.6 12.8 12 11.9 13 ...
 $ TAD_09      : num [1:352] 9.58 10.55 10.11 10.25 10.64 ...
 $ TAD_10      : num [1:352] 6.87 7.48 6.01 6.58 7.12 ...
 $ TAD_11      : num [1:352] 2.86 3.02 1.86 4.45 4.02 ...
 $ TAD_12      : num [1:352] 0.265 -0.36 0.527 0.46 1.023 ...
 $ TAmax_01    : num [1:352] 10.7 3 8.85 4.4 4.9 5.05 5.55 5.5 3.6 5.5 ...
 $ TAmax_02    : num [1:352] 3.7 6.4 6.7 7 9.6 5.45 5.55 7.65 5.3 8.8 ...
 $ TAmax_03    : num [1:352] 8.6 10.5 9.65 9.8 10.4 ...
 $ TAmax_04    : num [1:352] 10.7 13.1 14 12.2 13.6 ...
 $ TAmax_05    : num [1:352] 14.8 14.2 13.9 15.4 13.6 ...
 $ TAmax_06    : num [1:352] 15.4 15.9 16.2 15.2 16.2 ...
 $ TAmax_07    : num [1:352] 16.1 16.5 17.6 16.4 17.5 ...
> # Shift Month
> Weather_Area_Region <- data.table(Weather_Area_Region)
> Weather_Area_Region[, year := as.numeric(year)]
> str(Weather_Area_Region)
Classes 'data.table' and 'data.frame':  352 obs. of  77 variables:
 $ area_id     : chr  "4400000000" "4400000000" "4400000000" "4400000000" ...
 $ region_1    : chr  "충남" "충남" "충남" "충남" ...
 $ region_2    : chr  "서산" "서산" "서산" "서산" ...
 $ year        : num  2000 2001 2002 2003 2004 ...
 $ area        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ TAD_01      : num  -0.276 -1.948 0.145 -1.768 -0.934 ...
 $ TAD_02      : num  -1.286 -0.421 0.157 0.448 0.81 ...
 $ TAD_03      : num  1.93 1.69 2.92 2.28 2.43 ...
 $ TAD_04      : num  4.74 5.51 6.16 5.93 5.53 ...
 $ TAD_05      : num  7.51 8.74 8.14 9.11 8.3 ...
 $ TAD_06      : num  10.4 10.7 10.5 10.4 11 ...
 $ TAD_07      : num  12.5 12.7 12.3 11.8 12.4 ...
 $ TAD_08      : num  12.6 12.8 12 11.9 13 ...
 $ TAD_09      : num  9.58 10.55 10.11 10.25 10.64 ...
 $ TAD_10      : num  6.87 7.48 6.01 6.58 7.12 ...
```

3. 최종 데이터셋 완성(7/8)

```
> Weather_Area_Region <- as.data.frame(Weather_Area_Region)
> Weather_Area_Region  <- Weather_Area_Region[do.call("order", Weather_Area_Region[c("region_
> index <- word(names(Weather_Area_Region),-1, sep="_") %in% c('09','10','11','12')
> Weather_Area_Region[2:nrow(Weather_Area_Region),index] <- Weather_Area_Region[1:(nrow(Weath
> Weather_Area_Region <- as.data.table(Weather_Area_Region)
> rm(index)
>
> tmp <- Weather_Area_Region[, .SD[which.min(year)], by=c("region_1", "region_2"), .SDcols="y
> tmp[, remove := 1]
>
> Weather_Area_Region <- as.data.frame(Weather_Area_Region)
> tmp <- as.data.frame(tmp)
> Weather_Area_Region <- merge(Weather_Area_Region, tmp, by=c("region_1", "region_2", "year")
> rm(tmp)
> Weather_Area_Region <- subset(Weather_Area_Region, is.na(remove))
> Weather_Area_Region <- as.data.table(Weather_Area_Region)
> Weather_Area_Region[, remove := NULL]
> str(Weather_Area_Region)
Classes 'data.table' and 'data.frame':  330 obs. of  77 variables:
 $ region_1    : chr  "경남" "경남" "경남" "경남" ...
 $ region_2    : chr  "창녕" "창녕" "창녕" "창녕" ...
 $ year        : num  2001 2002 2003 2004 2005 ...
 $ area_id     : chr  "4800000000" "4800000000" "4800000000" "4800000000" ...
 $ area        : num  1121 1121 1121 1121 1121 ...
 $ TAD_01      : num  -0.485 0.673 -0.777 -0.478 -0.502 ...
 $ TAD_02      : num  0.6777 1 0.9121 1.2 0.0589 ...
 $ TAD_03      : num  2.57 3.17 2.55 2.78 2.08 ...
 $ TAD_04      : num  5.01 5.39 4.99 5.21 5.25 ...
 $ TAD_05      : num  7.13 6.73 6.82 7.01 6.96 ...
 $ TAD_06      : num  8.7 8.62 8.46 8.64 9.1 ...
> head(Weather_Area_Region, 5)
   region_1 region_2 year    area_id area    TAD_01    TAD_02 TAD_03 TAD_04
1:     경남     창녕 2001 4800000000 1121 -0.48508 0.677679 2.5718 5.0088
2:     경남     창녕 2002 4800000000 1121  0.67258 1.000000 3.1750 5.3863
3:     경남     창녕 2003 4800000000 1121 -0.77661 0.912054 2.5452 4.9887
4:     경남     창녕 2004 4800000000 1121 -0.47782 1.200000 2.7823 5.2125
5:     경남     창녕 2005 4800000000 1121 -0.50202 0.058929 2.0794 5.2450
   TAD_05 TAD_06  TAD_07 TAD_08 TAD_09 TAD_10 TAD_11  TAD_12 TAmax_01 TAmax_02
1: 7.1286 8.7037 10.1480 9.8335 7.7838 5.6468 2.6125 0.28427   4.0125   7.2375
2: 6.7325 8.6250  9.7200 9.4862 7.8725 5.7653 2.1488 0.28911   6.8250   6.1875
3: 6.8226 8.4625  8.7774 9.4138 7.6580 4.8063 1.7363 0.84875   4.8750   5.7750
4: 7.0077 8.6438 10.2411 9.6895 8.3150 5.0419 3.7400 0.53952   5.0625   7.9500
5: 6.9569 9.1025  9.8540 9.7234 8.0525 5.3395 2.9988 1.02581   4.5750   4.1250
   TAmax_03 TAmax_04 TAmax_05 TAmax_06 TAmax_07 TAmax_08 TAmax_09 TAmax_10
1:   9.1875   11.700   12.487   13.087   14.025   14.250   13.200   10.800
2:   8.7375   10.875   12.150   13.500   13.350   13.763   11.850   10.050
3:   7.9500   10.425   11.250   12.300   12.675   13.237   13.425   10.762
4:   9.4500   11.363   13.125   13.237   14.362   14.288   12.750   10.538
5:   8.4000   11.512   12.188   13.725   14.138   13.725   12.300   10.500
   TAmax_11 TAmax_12 TAmin_01 TAmin_02 TAmin_03 TAmin_04 TAmin_05 TAmin_06
1:   8.5875   6.4125  -5.3250  -3.5625  -2.8500  -1.6125    2.475   3.0375
2:   8.1000   5.5500  -4.8750  -3.5625  -2.6625   0.0375    1.950   3.9000
3:   8.1375   6.4500  -5.4000  -3.4500  -1.4250  -0.4875    1.050   4.6875
4:   9.6375   5.8875  -5.9625  -4.2000  -3.1875  -0.4125    2.100   3.9750
```

3. 최종 데이터셋 완성(8/8)

```
> # Weather_Area_Region and onion
> Weather_Area_Region[, year := as.integer(year)]
>
> Weather_Area_Region <- as.data.frame(Weather_Area_Region)
> onion <- as.data.frame(onion)
> Dataset <- merge(Weather_Area_Region, onion, by=c("region_1", "year"), all.x=TRUE, all.y=FA
> rm(Weather_Area_Region, onion)
>
> Dataset <- as.data.table(Dataset)
> Dataset <- subset(Dataset, !is.na(y.replace))
> Dataset[, y := NULL]
> setnames(Dataset, "y.replace", "y" )
> str(Dataset)
Classes 'data.table' and 'data.frame':  330 obs. of  78 variables:
 $ region_1    : chr  "경남" "경남" "경남" "경남" ...
 $ year        : int  2001 2001 2001 2002 2002 2002 2003 2003 2003 2004 ...
 $ region_2    : chr  "창녕" "함양" "합천" "창녕" ...
 $ area_id     : chr  "4800000000" "4800000000" "4800000000" "4800000000" ...
 $ area        : num  1121 665 1205 1121 665 ...
 $ TAD_01      : num  -0.485 -0.339 -0.465 0.673 0.369 ...
 $ TAD_02      : num  0.678 0.304 1.01 1 0.503 ...
 $ TAD_03      : num  2.57 1.3 2.49 3.17 1.8 ...
 $ TAD 04      : num  5.01 2.99 4.96 5.39 3.04 ...
> # Re-order columns
> names(Dataset)
 [1] "region_1"     "year"         "region_2"     "area_id"      "area"
 [6] "TAD_01"       "TAD_02"       "TAD_03"       "TAD_04"       "TAD_05"
[11] "TAD_06"       "TAD_07"       "TAD_08"       "TAD_09"       "TAD_10"
[16] "TAD_11"       "TAD_12"       "TAmax_01"     "TAmax_02"     "TAmax_03"
[21] "TAmax_04"     "TAmax_05"     "TAmax_06"     "TAmax_07"     "TAmax_08"
[26] "TAmax_09"     "TAmax_10"     "TAmax_11"     "TAmax_12"     "TAmin_01"
[31] "TAmin_02"     "TAmin_03"     "TAmin_04"     "TAmin_05"     "TAmin_06"
[36] "TAmin_07"     "TAmin_08"     "TAmin_09"     "TAmin_10"     "TAmin_11"
[41] "TAmin_12"     "DTD_01"       "DTD_02"       "DTD_03"       "DTD_04"
[46] "DTD_05"       "DTD_06"       "DTD_07"       "DTD_08"       "DTD_09"
[51] "DTD_10"       "DTD_11"       "DTD_12"       "RAINSUM_01"   "RAINSUM_02"
[56] "RAINSUM_03"   "RAINSUM_04"   "RAINSUM_05"   "RAINSUM_06"   "RAINSUM_07"
[61] "RAINSUM_08"   "RAINSUM_09"   "RAINSUM_10"   "RAINSUM_11"   "RAINSUM_12"
[66] "SUM_SS_HR_01" "SUM_SS_HR_02" "SUM_SS_HR_03" "SUM_SS_HR_04" "SUM_SS_HR_05"
[71] "SUM_SS_HR_06" "SUM_SS_HR_07" "SUM_SS_HR_08" "SUM_SS_HR_09" "SUM_SS_HR_10"
[76] "SUM_SS_HR_11" "SUM_SS_HR_12" "y"
> Dataset <- subset(Dataset, select=c(area_id, region_1, region_2, area, year, y, TAD_01:SUM_
> str(Dataset)
Classes 'data.table' and 'data.frame':  330 obs. of  78 variables:
 $ area_id     : chr  "4800000000" "4800000000" "4800000000" "4800000000" ...
 $ region_1    : chr  "경남" "경남" "경남" "경남" ...
 $ region_2    : chr  "창녕" "함양" "합천" "창녕" ...
 $ area        : num  1121 665 1205 1121 665 ...
 $ year        : int  2001 2001 2001 2002 2002 2002 2003 2003 2003 2004 ...
 $ y           : num  6045 6045 6045 6150 6150 ...
 $ TAD_01      : num  -0.485 -0.339 -0.465 0.673 0.369 ...
 $ TAD_02      : num  0.678 0.304 1.01 1 0.503 ...
 $ TAD_03      : num  2.57 1.3 2.49 3.17 1.8 ...
```

Ⅳ. 모형구축

1. 다중공선성 해결 및 변수 선택
2. 모형 구축

1. 다중공선성 해결 및 변수 선택 (1/4)

- **다중공선성 해결 및 변수 선택**

- 기상변수들간의 다중공선성을 해결하고 분석에 활용할 최종 변수를 선택

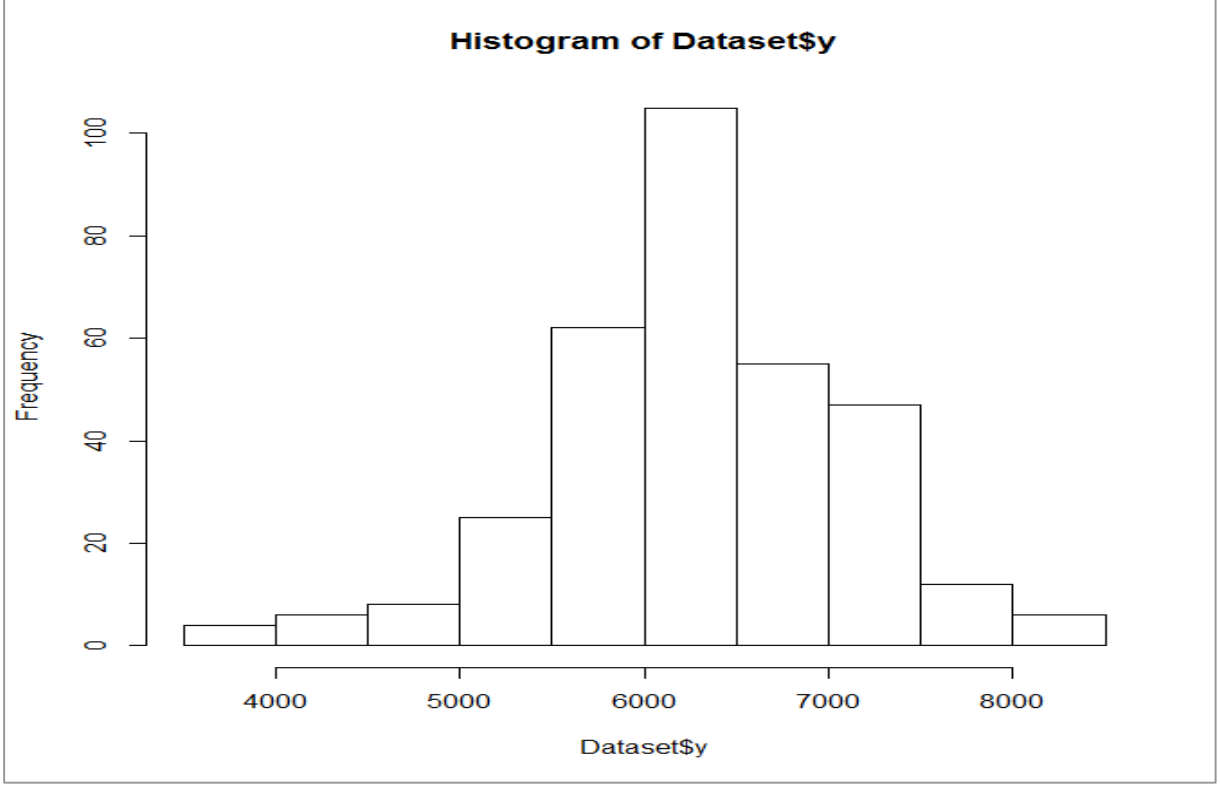
```
#=============================================================#
#Selecting Variables
#=============================================================#
#Checking Distribution
hist(Dataset$y)

#Creating Abs_Cor
Dataset <-as.data.frame(Dataset)
Abs_Cor <-abs(as.data.frame(cor(Dataset[,6:ncol(Dataset)])))
Abs_Cor$variable <-rownames(Abs_Cor)
str(Abs_Cor)

#Creating Select_Variables
Dataset <-as.h2o(Dataset)
Select_Variables <-h2o.glm(
y=6, x=7:ncol(Dataset),
training_frame = Dataset,
validation_frame = Dataset,
family = "gaussian",
link = "family_default",
lambda_search = TRUE,
fold_assignment = "AUTO")
Select_Variables@model$validation_metrics

#Checking RFresult
Imp_RF <-h2o.varimp(Select_Variables)
Imp_RF <-Imp_RF[order(-Imp_RF$coefficients),]
setnames(Imp_RF, "names", "variable")
str(Imp_RF)

#Remove multicollinearity
Select_Imp_RF = Imp_RF
j <-1
while(j <length(Select_Imp_RF$variable)) {
Temp <-subset(Abs_Cor, (Abs_Cor$variable!="y" &
Abs_Cor$variable!=Select_Imp_RF$variable[j] ), select=c("variable", Select_Imp_RF$variable[j]))
setnames(Temp, Select_Imp_RF$variable[j], "Pearson")
Temp <-Temp[order(-Temp$Pearson),]
Temp <-subset(Temp, Temp$Pearson >0.45 )
Select_Imp_RF <-merge(Select_Imp_RF, Temp, by="variable", all.x=TRUE, all.y=FALSE)
Select_Imp_RF <-subset(Select_Imp_RF, is.na(Pearson), select=-Pearson)
Select_Imp_RF <-as.data.frame(Select_Imp_RF[order(-Select_Imp_RF$coefficients),])
j <-j + 1
}
```

1. 다중공선성 해결 및 변수 선택 (2/4)

```
Select_Imp_RF <-subset(Select_Imp_RF, !is.na(coefficients))
str(Select_Imp_RF)
head(Select_Imp_RF)

rm(Imp_RF, Select_Variables, j, Temp)
```

- **hist()로 y값의 히스토그램을 그려 분포를 파악**
- **abs(cor())를 통해 변수들간의 상관계수 절대값을 산출**
- **h2o.glm()으로 모든 독립변수를 넣고 GLM분석 수행** (h2o.glm()함수 설명은 45p참고)
- **h2o.varimp()로 분석 모형의 변수 중요도 산출**
- **order()로 변수 중요도 순으로 내림차순 정렬**
- **while 반복문을 사용하여 상관성이 높은 독립변수들 중(0.45이상) 변수 중요도가 낮은 순으 로 반복해서 제거 (다중공선성이 높은 변수 제거)**
  - 특정 조건이 TRUE일때 블록 안의 문장을 반복 수행
    while(조건) {
    조건이 TRUE일때 수행할 문장 }
- **상관성이 높은 변수들 제거 후, 남은 변수들을 최종 변수로 선택**

> histogram 실행 결과



1. 다중공선성 해결 및 변수 선택 (3/4)

> 실행 결과

```
> # Creating Abs_Cor
> Dataset <- as.data.frame(Dataset)
> Abs_Cor <- abs(as.data.frame(cor(Dataset[,6:ncol(Dataset)])))
> Abs_Cor$variable <- rownames(Abs_Cor)
> str(Abs_Cor)
'data.frame':   73 obs. of  74 variables:
 $ y           : num  1 0.067 0.0744 0.0663 0.2067 ...
 $ TAD_01      : num  0.067 1 0.797 0.594 0.336 ...
 $ TAD_02      : num  0.0744 0.7966 1 0.8126 0.6083 ...
 $ TAD_03      : num  0.0663 0.5943 0.8126 1 0.9314 ...
 $ TAD_04      : num  0.207 0.336 0.608 0.931 1 ...
 $ TAD_05      : num  0.254 0.209 0.535 0.886 0.981 ...
 $ TAD_06      : num  0.281 0.164 0.494 0.86 0.969 ...
 $ TAD_07      : num  0.277 0.19 0.503 0.871 0.972 ...
 $ TAD_08      : num  0.298 0.21 0.522 0.874 0.966 ...
 $ TAD_09      : num  0.274 0.255 0.561 0.893 0.978 ...
 $ TAD_10      : num  0.248 0.384 0.661 0.935 0.972 ...
 $ TAD_11      : num  0.239 0.553 0.739 0.919 0.909 ...
 $ TAD_12      : num  0.00924 0.79787 0.85587 0.73437 0.5874 ...
> # Creating Select_Variables
> Dataset <- as.h2o(Dataset)
  |======================================================================| 100%
> Select_Variables <- h2o.glm(
+   y=6, x=7:ncol(Dataset),
+   training_frame = Dataset,
+   validation_frame = Dataset,
+   family = "gaussian",
+   link = "family_default",
+   lambda_search = TRUE,
+   fold_assignment = "AUTO")

  |======================================================================| 100%

> H2ORegressionMetrics: glm
** Reported on validation data. **

MSE:  517821
RMSE:  719.6
MAE:  557.95
RMSLE:  0.11731
Mean Residual Deviance :  517821
R^2 :  0.20683
Null Deviance :215441481
Null D.o.F. :329
Residual Deviance :170881061
Residual D.o.F. :259
AIC :5422.4
```

1. 다중공선성 해결 및 변수 선택 (4/4)

```
> # Checking RF result
> Imp_RF <- h2o.varimp(Select_Variables)
> Imp_RF <- Imp_RF[order(-Imp_RF$coefficients),]
> setnames(Imp_RF, "names", "variable")
> str(Imp_RF)
Classes 'H2OTable' and 'data.frame':    73 obs. of  3 variables:
 $ variable    : chr  "TAD_02" "TAD_03" "SUM_SS_HR_06" "RAINSUM_01" ...
 $ coefficients: num  38.1 30.6 29.8 29.5 27.2 ...
 $ sign        : chr  "POS" "POS" "NEG" "NEG" ...
 - attr(*, "header")= chr "Standardized Coefficient Magnitudes"
 - attr(*, "formats")= chr  "%s" "%5f" "%s"
 - attr(*, "description")= chr "standardized coefficient magnitudes"
>
> # Remove multicollinearity
> Select_Imp_RF = Imp_RF
> j <- 1
> while(j < length(Select_Imp_RF$variable)) {
+   Temp <- subset(Abs_Cor, (Abs_Cor$variable!="y" & Abs_Cor$variable!=Select_Imp_RF$variable[j] ), select=c("v
+   setnames(Temp, Select_Imp_RF$variable[j], "Pearson")
+   Temp <- Temp[order(-Temp$Pearson),]
+   Temp <- subset(Temp, Temp$Pearson > 0.45 )
+   Select_Imp_RF <- merge(Select_Imp_RF, Temp, by="variable", all.x=TRUE, all.y=FALSE)
+   Select_Imp_RF <- subset(Select_Imp_RF, is.na(Pearson), select=-Pearson)
+   Select_Imp_RF <- as.data.frame(Select_Imp_RF[order(-Select_Imp_RF$coefficients),])
+   j <- j + 1
+ }
> Select_Imp_RF <- subset(Select_Imp_RF, !is.na(coefficients))
> str(Select_Imp_RF)
'data.frame':   3 obs. of  3 variables:
 $ variable    : chr  "TAD_02" "SUM_SS_HR_06" "RAINSUM_09"
 $ coefficients: num  38.1 29.8 12
 $ sign        : chr  "POS" "NEG" "POS"
> head(Select_Imp_RF)
      variable coefficients sign
4       TAD_02       38.069  POS
3 SUM_SS_HR_06       29.830  NEG
2   RAINSUM_09       11.985  POS
```

2. 모형 구축(1/2)

- **GLM 모형구축**

- 최종 선택 변수를 활용하여 GLM 모형을구축

```
#=================================================================#
#Creating Model
#=================================================================#
Dataset <-as.data.table(Dataset)
Train <-subset(Dataset, select=c("y", Select_Imp_RF$variable))
Train <-as.h2o(Train)

#Created_Model
Created_Model <-h2o.glm(
y=1, x=2:ncol(Train),
training_frame = Train,
family = "gaussian",
link = "family_default",
lambda_search = TRUE,
fold_assignment = "AUTO")
Created_Model
rm(Train)

#Check
ls()
tables()
gc()
#=================================================================#
```

- **subset()으로 최종 선택 변수와 종속 변수를 선택하여 트레이닝 데이터 생성**
- **as.h2o()로 트레이닝 데이터를 h2o 프레임 형태로변환**
- **h2o.glm()으로 트레이닝 데이터를 학습하여 GLM 모형구축**
  - family : y값이 정규분포 형태이기 때문에 “Gaussian” 설정
  - link : family인 Gaussian에 종속되는 기본 값 설정
  - lambda_search : 모형의 람다값을 산출
  - fold_assignment : 교차 검증 구조 “AUTO” 설정

2. 모형 구축(2/2)

> 실행 결과

```
> #=================================================================#
> # Creating Model
> #=================================================================#
> Dataset <- as.data.table(Dataset)
> Train <- subset(Dataset, select=c("y", Select_Imp_RF$variable))
> Train <- as.h2o(Train)
  |=================================================================================| 100%
>
> # Created_Model
> Created_Model <- h2o.glm(
+   y=1, x=2:ncol(Train),
+   training_frame = Train,
+   family = "gaussian",
+   link = "family_default",
+   lambda_search = TRUE,
+   fold_assignment = "AUTO")
  |=================================================================================| 100%
> Created_Model
Model Details:
==============

H2ORegressionModel: glm
Model ID:  GLM_model_R_1478757482809_25
GLM Model: summary
    family     link                                regularization
1 gaussian identity Elastic Net (alpha = 0.5, lambda = 0.05798 )
                                                   lambda_search
1 nlambda = 100, lambda.max = 579.8, lambda.min = 0.05798, lambda.1se = -1.0
  number_of_predictors_total number_of_active_predictors number_of_iterations
1                          3                            3                    0
  training_frame
1          Train

Coefficients: glm coefficients
        names coefficients standardized_coefficients
1   Intercept  6670.998083               6292.359091
2      TAD_02   162.606955                188.095358
3 SUM_SS_HR_06  -342.717716               -366.024551
4   RAINSUM_09     0.622781                 32.737116

H2ORegressionMetrics: glm
** Reported on training data. **

MSE:  532566
RMSE:  729.77
MAE:  574.39
RMSLE:  0.11764
Mean Residual Deviance :  532566
R^2 :  0.18425
Null Deviance :215441481
Null D.o.F. :329
Residual Deviance :175746725
Residual D.o.F. :326
AIC :5297.7
```

Ⅴ. 모형검증

1. 교차 검증

1. 교차검증(1/3)

- **교차 검증**

- 구축한 GLM 모형의 교차 검증(k-fold Cross-validation)수행

```
#============================================================#
#k-fold Cross-vaildation
#============================================================#
#GLM_Validation_Result
GLM_Validation_Result <-list()

for(Signifi_Year in min(Dataset$year):max(Dataset$year)){
 j <-Signifi_Year-min(Dataset$year)+1
  #Setting Train and Test
 Train <-subset(Dataset, year!=Signifi_Year, select=c("y", Select_Imp_RF$variable))
 str(Train)
 Train <-as.h2o(Train)
 Test_Y <-subset(Dataset, year==Signifi_Year, select=c("y"))
 Test_X <-subset(Dataset, year==Signifi_Year, select=c(Select_Imp_RF$variable))
 Test_X <-as.h2o(Test_X)
 #Created_Model
 GLM_Validation <-h2o.glm(
                    y=1, x=2:ncol(Train),
                    training_frame =Train,
                    family = "gaussian",
                    link ="family_default",
                    fold_assignment ="AUTO",
                    lambda =0.04849)
 #print
 print(paste0("Year=", Signifi_Year," ing..."))
 print(GLM_Validation)

 Prediction <-as.data.table(h2o.predict(object=GLM_Validation, newdata=Test_X))
 Prediction <-cbind(Prediction, Test_Y)
 Prediction[, MAPE :=abs((y-predict)/y)*100]

 validation_metrics <-data.frame(Year = Signifi_Year,
 R2 = GLM_Validation@model$training_metrics@metrics$r2,
 MAPE = mean(Prediction$MAPE,na.rm=TRUE))

 GLM_Validation_Result[[j]] <-validation_metrics
}

GLM_Validation_Result_list <-rbindlist(GLM_Validation_Result)
mean(GLM_Validation_Result_list$MAPE)
mean(GLM_Validation_Result_list$R2)
#============================================================#
```

1. 교차 검증(2/3)

- list()로 연도별 교차검증 결과를 저장할 리스트 생성
- for 반복문으로 연도별 교차 검증을 반복 수행
- Signifi_Year 변수에 검증할 연도를 차례대로 입력
- Signifi_Year 연도를 제외한 나머지 연도의 데이터를 Train 데이터로 생성
- Signifi_Year 연도의 데이터를 Test 데이터로 생성
- Train데이터를 사용하여 h2o.glm()으로 GLM 모형구축
- h2o.predict()로 Test 데이터를 활용하여 예측값 산출
- cbind()로 예측값과 실제y값 병합
- R2 및 MAPE 산출
- for반복문이 끝난 후, rbindlist()로 교차검증 결과 병합
- mean()으로 평균 MAPE와 R2 계산

1. 교차검증(3/3)

> 실행 결과

```
> #============================================================#
> # k-fold Cross-vaildation
> #============================================================#
> # GLM_Validation_Result
> GLM_Validation_Result <- list()
>
> for(Signifi_Year in min(Dataset$year):max(Dataset$year)){
+   j <- Signifi_Year-min(Dataset$year)+1
+   # Setting Train and Test
+   Train <- subset(Dataset, year!=Signifi_Year, select=c("y", Select_Imp_RF$variable))
+   str(Train)
+   Train <- as.h2o(Train)
+   Test_Y <- subset(Dataset, year==Signifi_Year, select=c("y"))
+   Test_X <- subset(Dataset, year==Signifi_Year, select=c(Select_Imp_RF$variable))
+   Test_X <- as.h2o(Test_X)
+   # Created_Model
+   GLM_Validation <- h2o.glm(
+       y=1, x=2:ncol(Train),
+       training_frame = Train,
+       family = "gaussian",
+       family = "gaussian",
+       link = "family_default",
+       fold_assignment = "AUTO",
+       lambda = 0.04849)
+   # print
+   print(paste0("Year=", Signifi_Year," ing..."))
+   print(GLM_Validation)
+
+   Prediction <- as.data.table(h2o.predict(object=GLM_Validation, newdata=Test_X))
+   Prediction <- cbind(Prediction, Test_Y)
+   Prediction[, MAPE := abs((y-predict)/y)*100]
+
+   validation_metrics <- data.frame(Year = Signifi_Year,
+                                    R2 = GLM_Validation@model$training_metrics@metrics$r2,
+                                    MAPE = mean(Prediction$MAPE, na.rm=TRUE))
+
+   GLM_Validation_Result[[j]] <- validation_metrics
+ }
Classes 'data.table' and 'data.frame':  308 obs. of  4 variables:
 $ y           : num  6150 6150 6150 5883 5883 ...
 $ TAD_02      : num  1 0.503 1.072 1.379 0.912 ...
 $ SUM_SS_HR_06: num  3.02 1.79 2.86 1.96 2.25 ...
 $ RAINSUM_09  : num  43.5 19.6 43.1 65.5 39 ...
 - attr(*, ".internal.selfref")=<externalptr>
 |======================================================================| 100%
> GLM_Validation_Result_list <- rbindlist(GLM_Validation_Result)
> mean(GLM_Validation_Result_list$MAPE)
[1] 9.735
> mean(GLM_Validation_Result_list$R2)
[1] 0.18579
```

본 문서의 내용은 기상청의 기상기후 빅데이터 분석(https://bd.kma.go.kr)의
분석 플랫폼 활용을 위한 R 프로그래밍 교육 자료 입니다.