|  |
| --- |
| **데이터사이언스기초 6주차 실습과제** |
| **실습 #2 보고서** |

|  |  |
| --- | --- |
| **이름** | 안지용 |
| **학번** | 20177140 |
| **소속**  **학과/대학** | 소프트웨어융합대학 콘텐츠IT |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
* SMARTLEAD 제출 데드라인: 2021. 4. 06(화요일) ~ 4.12(월요일) 23:59 까지
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
  + 예외 없음
* SMARTLEAD에 아래의 파일을 제출 해 주세요
  + 보고서 형식으로 제출
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.

## <실습 과제>

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Q 1] switch(조건문), for/while/repeat(반복문) 사용  <사용되는 데이터 : Pre\_Season\_Batter.csv>  Pre\_Season\_Batter.csv의 데이터를 이용하여 조건식을 적용하려고 한다. 이 Pre\_Season\_Batter 에는 총 1,393 개의 행과 29개의 열로 구성되어 있다. 그중 “team”열을 이용하여, 팀의 연고지를 파악하려고 한다. 팀의 연고지를 파악하여 새로운 열인 “hometown”로 생성한다.  **Ex) teme가 LG 인 경우 “hometown”의 값은 서울, 삼성인 경우 “hometown”의 값은 대구로 설정한다.**  **<“team” 별 연고지(hometown)>**   |  |  |  |  | | --- | --- | --- | --- | | **team** | **hometown** | **team** | **hometown** | | **LG** | **서울** | **한화** | **대전** | | **NC** | **창원** | **넥센** | **서울** | | **롯데** | **부산** | **삼성** | **대구** | | **KT** | **수원** | **SK** | **인천** | | **KIA** | **광주** | **두산** | **서울** | | **히어로즈** | **서울** | **우리** | **서울** | | **현대** | **수원** |  |  |   **<진행사항>**   1. **Read.csv를 사용하고, header = T, stringsAsFactors = F로 줄 것** 2. **반드시 함수를 이용하여 진행하되 매개 변수의 수는 자유롭게 지정한다. 또한 반환값(return)은 hometown 열이 추가된 데이터프레임으로 정한다.** 3. **unique()를 이용하여 중복되지 않은 값을 추출한다.** 4. **함수의 이름은 team2hometown로 설정한다.** 5. **조건문을 반드시 쓰고(switch 사용 추천), for 또는 while을 사용할 것(apply 사용하지 말 것)**   **<출력 예시>**  **(1)**    **(2)** |

답변:

1. **unique(데이터프레임$team) 출력 결과(10 점)**



1. **team2hometown 함수 코드 및str(team2hometown(데이터프레임)) 실행 결과(40 점)**

#team2hometown 함수 작성

team2hometown=function(df){

  y=c()

  for(i in df$team){

    if(i=="LG"){

      y=c(y,"서울")

    }else if(i=="NC"){

      y=c(y,"창원")

    }else if(i=="롯데"){

      y=c(y,"부산")

    }else if(i=="KT"){

      y=c(y,"수원")

    }else if(i=="KIA"){

      y=c(y,"광주")

    }else if(i=="히어로즈"){

      y=c(y,"서울")

    }else if(i=="현대"){

      y=c(y,"수원")

    }else if(i=="한화"){

      y=c(y,"대전")

    }else if(i=="넥센"){

      y=c(y,"서울")

    }else if(i=="삼성"){

      y=c(y,"대구")

    }else if(i=="SK"){

      y=c(y,"인천")

    }else if(i=="두산"){

      y=c(y,"서울")

    }else if(i=="우리"){

      y=c(y,"서울")

    }

  }

  df$hometown=y ##df에 hometown 속성 추가

  return(df)  ##hometown열이 추가된 df 리턴

}

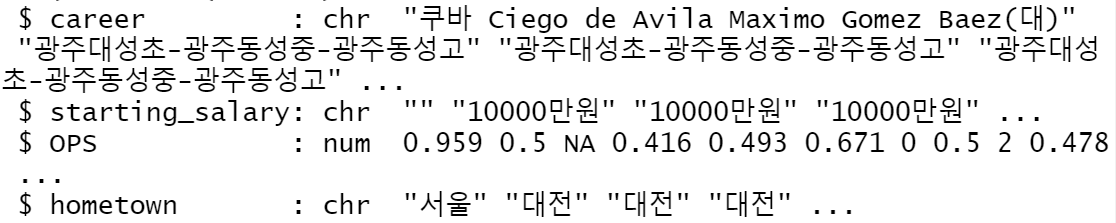
df = read.csv("Pre\_Season\_Batter.csv",header=T,stringsAsFactors = F)

DF=team2hometown(df)

unique(DF$team)

str(team2hometown(DF))

**실행결과**



소스 파일 같이 제출

파일 이름 : project1.R

## <실습 과제>

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Q 2] **Boxplot 그리기!!**  **<사용되는 데이터 - Pre\_Season\_Batter.csv, Regular\_Season\_Batter.csv>**  **Pre\_Season\_Batter.csv 과 Regular\_Season\_Batter.csv 데이터의 이상값(outlier)을 비교하기 위해 Boxplot을 그리려고 한다. 문제에서 사용되는 열(Column)은 AB, R, H, E, SLG, OBP로 총 6개이고, 아래의 표에 정리되어있다. outlier를 삭제하는 함수(removeOutliers)를 작성하여 비교를 진행한다.**  **<Pre\_Season\_Batter.csv와 Regular\_Season\_Batter.csv 정보>**   * **Regular\_Season\_Batter.csv : KBO에서 활약한 타자들의 역대 정규시즌 성적을 포함하여 몸무게, 키 ,생년월일 등의 기본정보** * **Pre\_Season\_Batter.csv : KBO에서 활약한 타자들의 역대 시범경기(정규시즌 직전에 여는 연습경기) 성적** * **Pre\_Season\_Batter와 Regular\_Season\_Batter의 사용되는 열(Column) 정보**  |  |  |  |  | | --- | --- | --- | --- | | **AB** | **타수** | **R** | **득점** | | **H** | **안타** | **E** | **실책** | | **SLG** | **장타율** | **OBP** | **출루율** |   **<참고사항>**   1. **data.table::fread를 사용하여 Pre\_Season\_Batter.csv와 Regular\_Season\_Batter.csv를 불러온다(Window환경에서는 encoding = “UTF-8”을 줄 것)** 2. **removeOutliers 함수**  * **매개변수(argument)로는 데이터프레임으로 받고 각 열의 데이터형은 숫자형(integer or numeric)으로 제한한다.** * **반환(return)은 outlier가 제거된 데이터프레임으로 한다.** * **quantile, IQR 함수를 사용하며, na.rm 은 TRUE 로 한다(na.rm = T)** * **lower / up fence를 계산하여 outlier 를 제거한다(아래의 공식 사용)**   **lower fence = Q1 – (1.5 \* IQR) up fence = Q3 + (1.5 \* IQR)**   * **outlier는 lower fence 미만이거나 up fence 를 초과할 경우를 말한다(OR)** * **반복문 또는 apply 계열 함수를 사용할 수 있고, 각 열의 길이(length)가 다를 경우 outlier에 해당하는 자리에NA를 삽입하여 길이를 맞춘다.**   **<출력예시> - removeOutlier로 outlier가 제거 된 후 boxplot을 적용하면,**  **boxplot자체에서 다시 outlier를 계산하기 때문에 아래와 같이 outlier가 max 근처에 도포될 수 있음!**  **하지만, 우리는 이미 outlier를 제거했으므로 그래프의 형태를 확인하자**  **(2)**    **(3)** |

답변:

1. **removeOutliers 함수 코드 첨부(중요하다 생각하는 부분에 주석 달 것)(60 점)**

removeOutliers=function(db){

  a1=IQR(db$AB,na.rm=T) #a1 변수에 AB컬럼 IQR 저장

  a2=IQR(db$R,na.rm=T)  #a2 변수에 R컬럼 IQR 저장

  a3=IQR(db$H,na.rm=T)  #a3 변수에 H컬럼 IQR 저장

  a4=IQR(db$E,na.rm=T)  #a4 변수에 E컬럼 IQR 저장

  a5=IQR(db$SLG,na.rm=T)#a5 변수에 SLG컬럼 IQR 저장

  a6=IQR(db$OBP,na.rm=T)#a6 변수에 OBP컬럼 IQR 저장

  #반복문을 써서 해결 하려고 했지만 오류가 나서

  #각 속성마다 따로 따로 계산을 하여 outlier를 제거했습니다.

  #AB속성

  range=c()

  #AB 속성의 boxplot을 range 벡터에 저장

  range=boxplot(db$AB)$stats

  lowerFence=range[2]-a1\*1.5#lowerfence 구하기

  upFence=range[4]+a1\*1.5#upfence 구하기

  #outlier 요소들을 NA로 바꿔주는 코드

  db$AB=ifelse(db$AB<lowerFence | db$AB>upFence,NA,db$AB)

  #R속성

  range=c()

  range=boxplot(db$R)$stats #boxplot 구하기

  lowerFence=range[2]-a2\*1.5#lowerfence 구하기

  upFence=range[4]+a2\*1.5#upfence 구하기

  #outlier 요소들을 NA로 바꿔주는 코드

  db$R=ifelse(db$R<lowerFence | db$R>upFence,NA,db$R)

  boxplot(db$R)$stats

  #H속성 다른 속성들도 위와 같은 작업으로 진행하였습니다.

  range=c()

  range=boxplot(db$H)$stats

  lowerFence=range[2]-a3\*1.5

  upFence=range[4]+a3\*1.5

  db$H=ifelse(db$H<lowerFence | db$H>upFence,NA,db$H)

  boxplot(db$H)$stats

  #E 속성

  range=c()

  range=boxplot(db$E)$stats

  lowerFence=range[2]-a4\*1.5

  upFence=range[4]+a4\*1.5

  db$E=ifelse(db$E<lowerFence | db$E>upFence,NA,db$E)

  boxplot(db$E)$stats

  #SLG 속성

  range=c()

  range=boxplot(db$SLG)$stats

  lowerFence=range[2]-a5\*1.5

  upFence=range[4]+a5\*1.5

  db$SLG=ifelse(db$SLG<lowerFence | db$SLG>upFence,NA,db$SLG)

  boxplot(db$SLG)$stats

  #OBP 속성

  range=c()

  range=boxplot(db$OBP)$stats

  lowerFence=range[2]-a6\*1.5

  upFence=range[4]+a6\*1.5

  db$OBP=ifelse(db$OBP<lowerFence | db$OBP>upFence,NA,db$OBP)

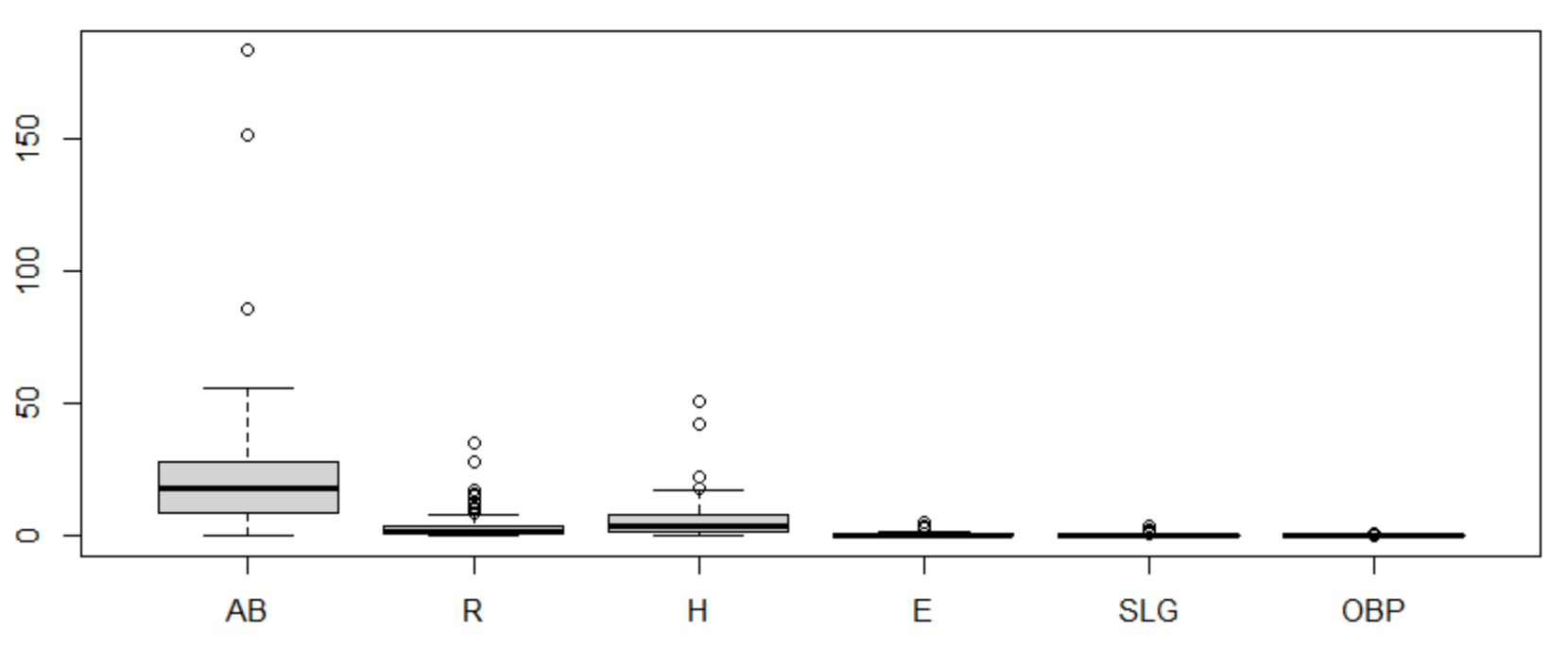
  boxplot(db$OBP)$stats

  return(db)

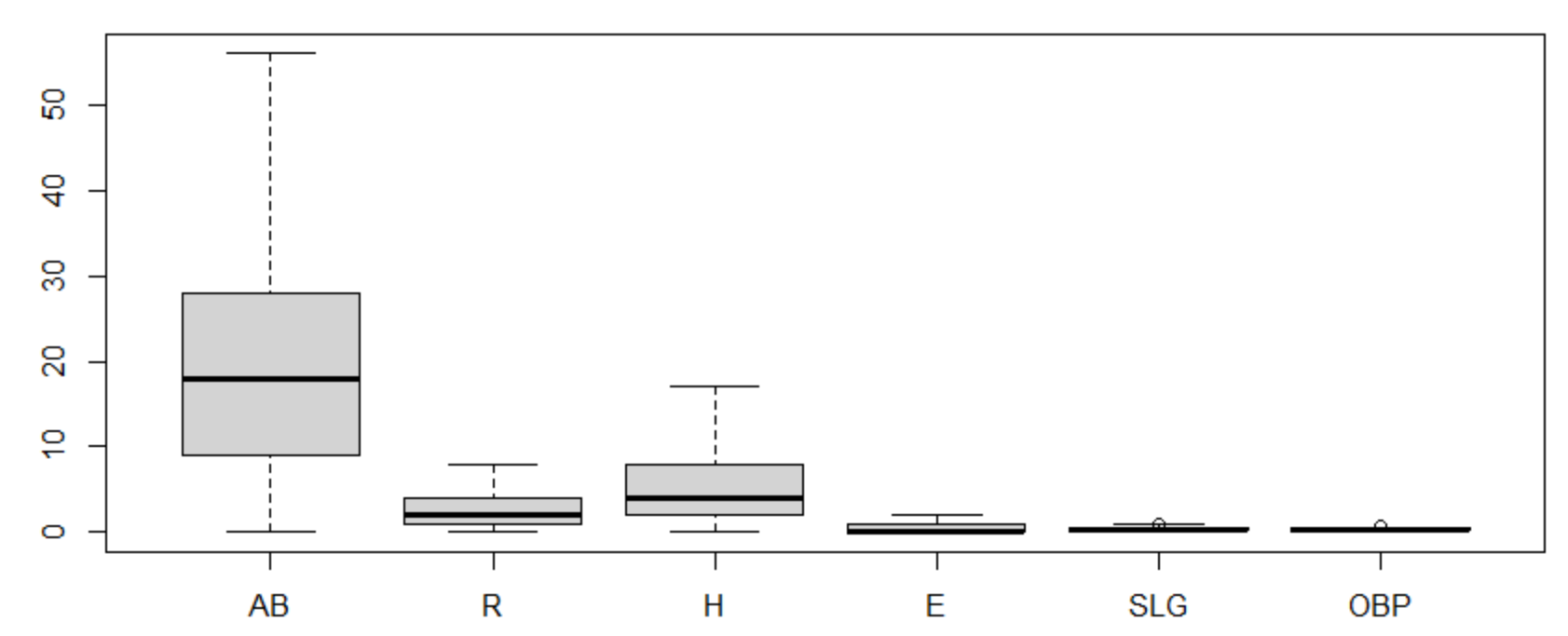
}

1. **Pre\_Season\_Batter.csv 의 outlier 제거 전/후 결과 첨부(20 점)**

**제거 전**

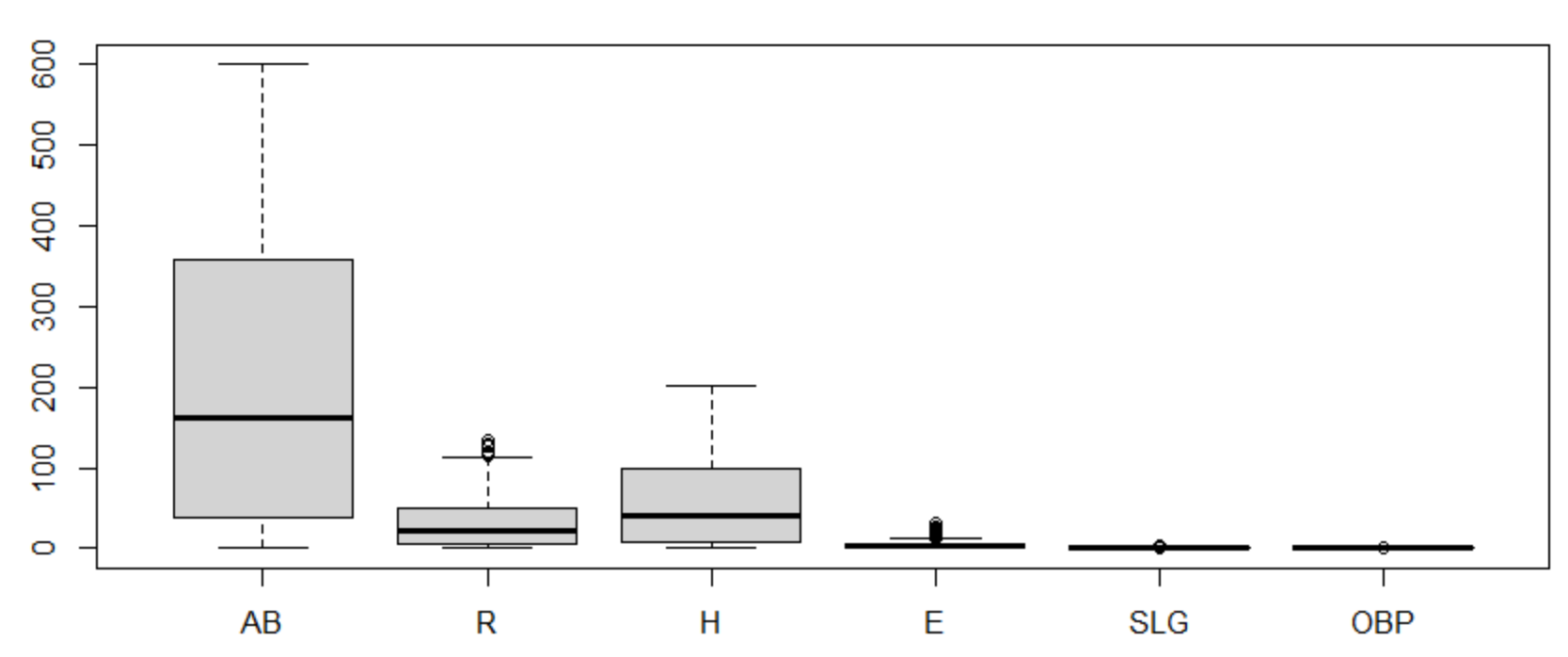


**제거 후**

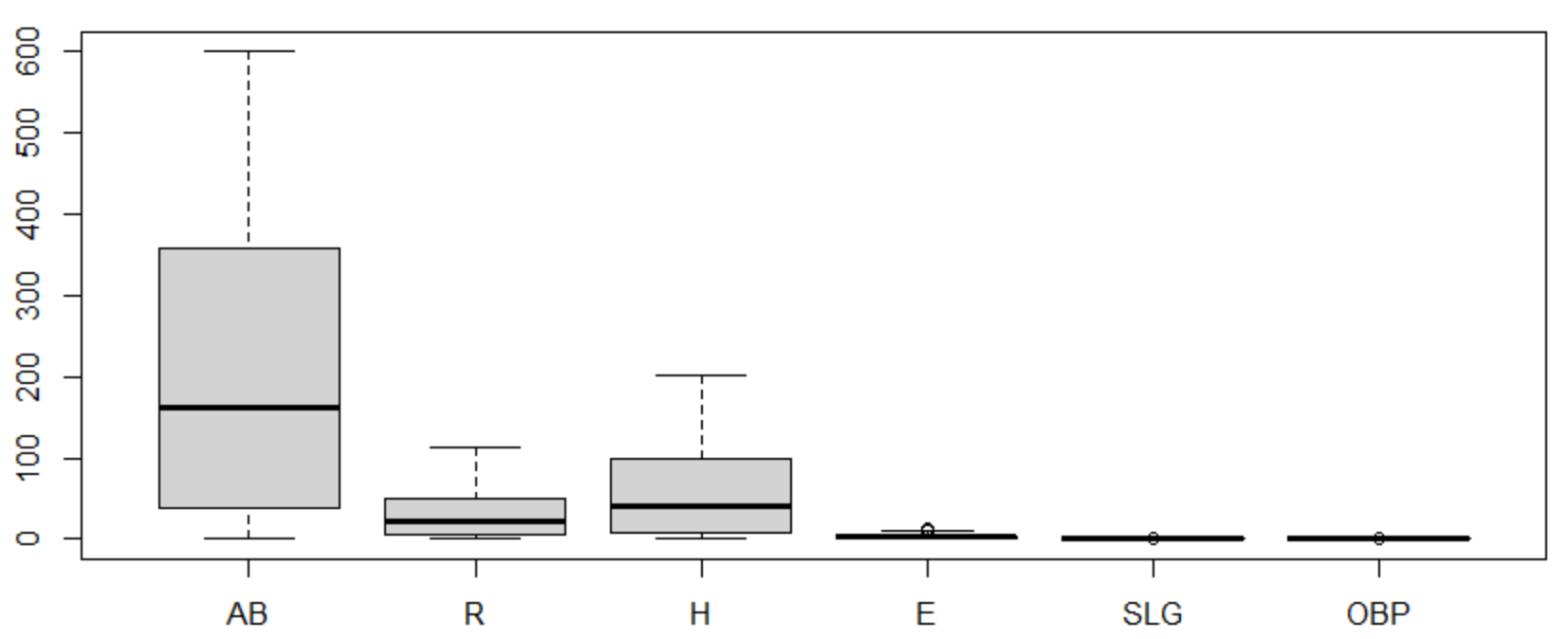


1. **Regular\_Season\_Batter.csv의 outlier 제거 전/후 결과 첨부(20 점)**

**제거 전**



**제거 후**



소스 파일 같이 제출

파일 이름 : project2.R

<실습 과제>

|  |
| --- |
| [Q 3] **groupby, summarise**  **<사용되는 데이터 - Pre\_Season\_Batter.csv >**  **출력예시처럼 group\_by와 summarise를 적절하게 적용하여 결과를 도출하시오.**  **<참고사항>**   * **DF는 fread 또는 read.csv를 이용하여 Pre\_Season\_Batter.csv 데이터프레임으로 변환할 것.!** * **사용되는 열(column)은 team, batter\_name, G, HR, X3B, X2B, H열** * **groub\_by, summarise, 사용**   **<출력 예시>** |

**답변: 코드와 출력 결과를 요약하여 첨부하세요**

1. **코드 (30점)**

library(tidyverse)

library(data.table)

df = read.csv("Pre\_Season\_Batter.csv",header=T,stringsAsFactors = F)

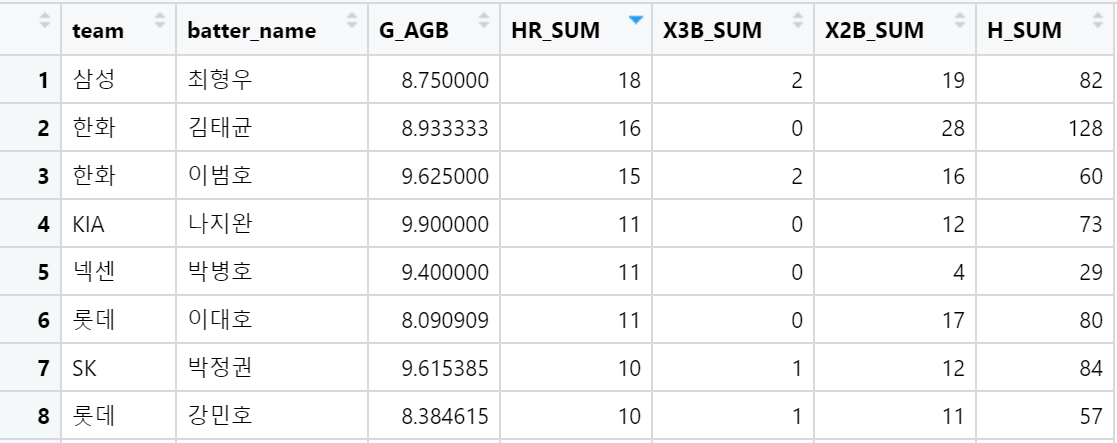
df=df[,c("team","batter\_name","G","HR","X3B","X2B","H")]

df2=summarise(group\_by(df,team,batter\_name),G\_AGB=mean(G),HR\_SUM=sum(HR),

              X3B\_SUM=sum(X3B),X2B\_SUM=sum(X2B),H\_SUM=sum(H))

View(df2)

1. **출력 결과 (20점)**



소스 파일 같이 제출

파일 이름 : project3.R