EDA 4장 과제

주의사항을 숙지하였고 모든 책임을 지겠습니다.

2019122041 송유진

---

1. 인터넷 검색 또는 출판 자료에서 가장 최근의 인구주택총조사 자료를 구하여 81쪽의 상자그림을 그리고, 81쪽의 상자 그림과 비교하여 구별 인구의 변동 추이를 설명하여라.
서울과 부산의 2000년과 최근 조사자료의 skewness, kurtosis를 계산하여 년도별, 도시별 비교하여라.
*** 국가통계사이트 KOSIS 검색해 보아라.**

```
setwd("D:\\2022-1(3-2)\\2022-01_탐자분\\R을 활용한 탐색적
자료분석")
pop2000<-read.csv("광역시-구 인구_2000.csv", head=T)
pop2020<-read.csv("광역시-구 인구_2020.csv", head=T)

head(pop2000)
tail(pop2000)
head(pop2020)
tail(pop2020)

attach(pop2000)
attach(pop2020)
```

2000년 인구주택총조사 자료를 pop2000으로 데이터프레임화 하였고 2020년 인구주택총조사 자료를 pop2020으로 데이터프레임화 하였다. 분석의 용이성을 위해 attach() 함수를 사용해주었다.

2000년

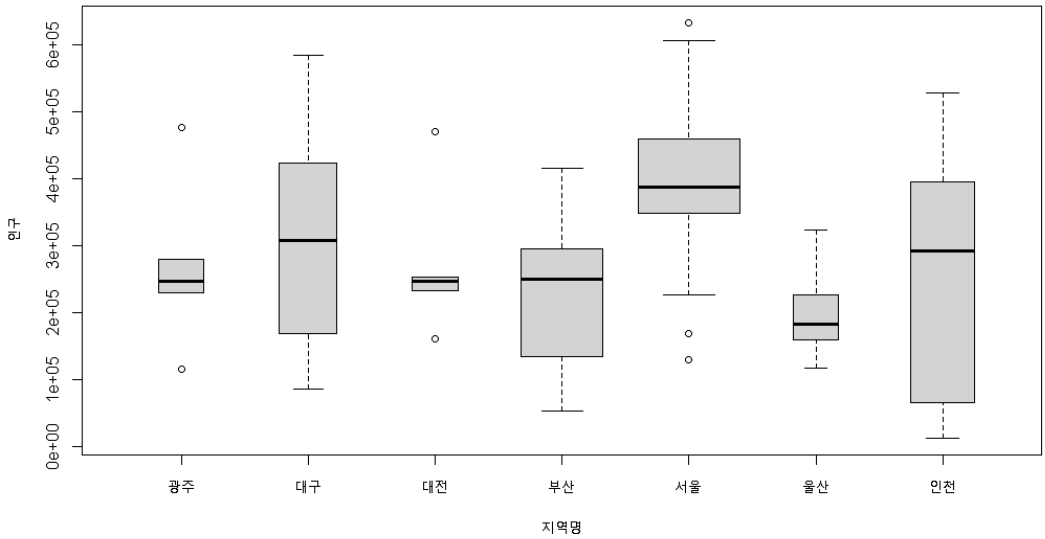
```
boxplot(인구~지역명, data=pop2000, varwidth=TRUE)
```

2020년

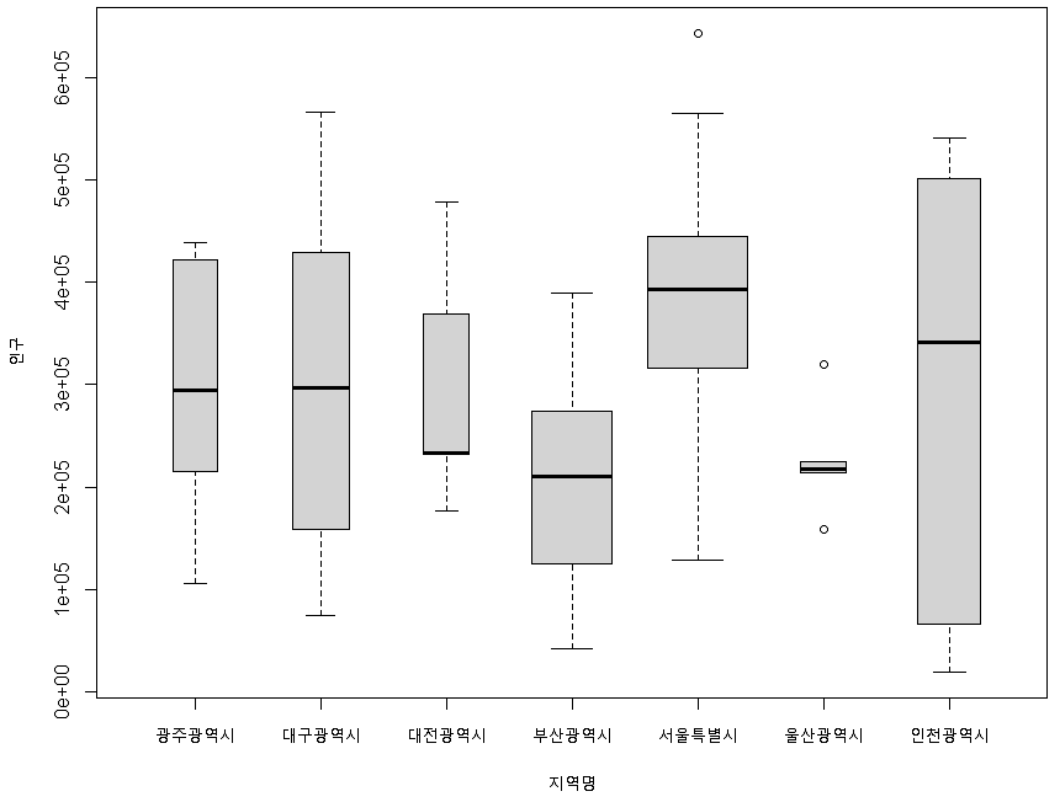
```
boxplot(인구~지역명, data=pop2020, varwidth=TRUE)
```

상단의 코드를 활용하여 boxplot을 그리고 그 결과는 아래와 다음과 같다.
varwidth를 지정하지 않은 boxplot은 해당 범주(ex. 서울, 광주 등)에 얼마나 많은 관측값이 있는지 알기 어렵다. 따라서 varwidth =TRUE를 통해 박스 플롯의 너비를 개수와 비례하도록 설정해주었다.

2000년



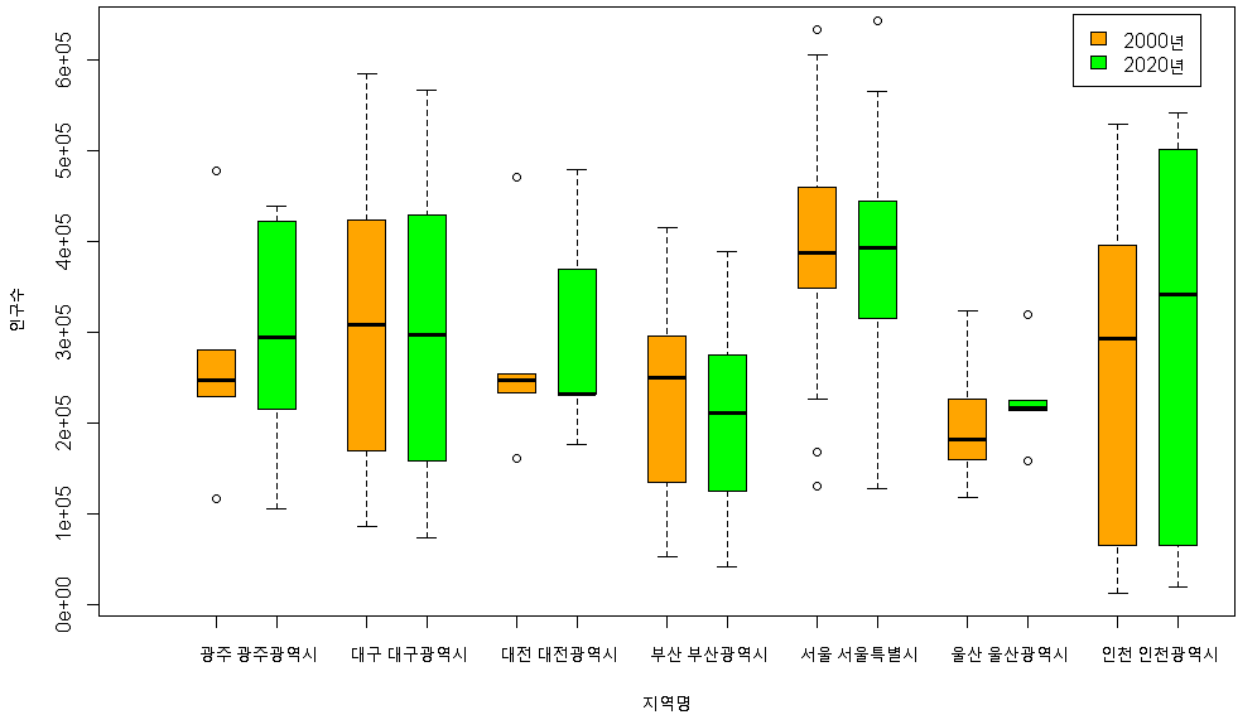
2020년



서울특별시는 총 25개의 구로 이루어져 있고 부산광역시는 16개, 대구광역시는 8개, 인천광역시는 10개, 광주광역시, 대전광역시, 울산광역시는 5개의 구로 이루어져 있다.
boxplot의 너비를 살펴보면 서울과 부산이 두드러지게 두꺼운 boxplot의 형태를 띠고 있음을 알 수 있다.

2000년과 2020년 인구의 변동 추이를 더 한 눈에 살펴보기 위해 아래의 코드를 실행하여 결과를 살펴보았다.

```
boxplot(인구~지역명, data=pop2000,
        xlab="지역명",ylab="인구수",col="orange",
        boxwex = 0.25, at = 1:7 - 0.2)

boxplot(인구~지역명, data=pop2020,
        xlab="지역명",ylab="인구수",col="green",
        boxwex = 0.25, at = 1:7 + 0.2,add=TRUE)

legend(6.5, 6.5e+5, c("2000년", "2020년"), fill = c("orange", "green"))
```



[광주]

광주의 경우 2000년에는 boxplot의 수염이 보이지 않고 outlier만 2개 존재하고 있다. 큰 outlier는 북구(477591명), 작은 outlier는 동구(116332명)이다. 2000년에 광주 인구수는 1350948명, 2020년 광주 인구수는 1477573명으로 증가했다. 5개의 구 중 광산구의 경우 2000년(247031명)에서 2020년(422502명)으로 크게 증가했기 때문에 광주의 인구수 또한 증가했음을 알 수 있다. 2020년에는 2000년에 관측되었던 outlier 2개가 사라지고 수염이 생겼음을 알 수 있다.

[대구]
대구의 경우 2000년 인구는 2473990명이고 2020년에는 2410700명으로 약간 감소하였다. 대구의 경우 2000년과 2020년 모두 outlier는 보이지 않는다. Boxplot을 살펴보았을 때 상자의 길이가 조금 더 길어졌음을 확인할 수 있다.

[대전]
대전은 광주와 비슷하게 2000년에 2개의 outlier들이 존재하고 있고 2020년에는 outlier가 사라지고 boxplot의 수염이 생겼음을 알 수 있다. 2000년에 outlier들은 서구(470327명), 유성구(161591명)이다. 2020년에 서구와 유성구는 각각 478629명, 368895명으로 증가한다. 유성구의 인구가 크게 증가하였고 이로 인해 대전 전체의 인구수 또한 2000년(1365961명)에서 2020년(1488435명)으로 증가했음을 알 수 있다.
대전은 5개의 구로 이루어져 있기 때문에 2020년에 2000년에 비해 boxplot 상자의 길이가 길어졌지만 Q1과 median이 거의 동일하게 형성되었음을 확인할 수 있다.

[부산]
부산의 경우 2000년과 2020년 모두 outlier 값은 관측되지 않고 있다. 2000년에 비해 2020년에 median 값이 내려갔음을 알 수 있다. 부산의 인구 또한 3655437명(2000년)에서 3349016명(2020년)으로 감소하였다.

[서울]
서울의 경우 outlier에서 변화가 있다. 2000년에는 outlier가 3개임을 알 수 있고 2020년에는 1개만 존재하고 있다. 2000년 서울의 boxplot을 벗어나는 outlier 중 작은 outlier들은 각각 종로구(168879명), 중구(130370명)이다. 2000년 서울의 큰 outlier는 송파구(632983명)이고 2020년 서울의 큰 outlier 또한 송파구(643288명)이다. 별 차이 없는 값이지만, 전체적으로 인구가 감소하여 box의 수염 안으로 들어오게 되었다. 데이터를 통해 서울 인구가 2000년에는 9853972명이었고 2020년에는 9586195명으로 감소했음을 알 수 있는데 이는 boxplot 수염의 길이가 2000년에 비해 2020년에 더 길어진 것을 통해 확인할 수 있다. Box의 길이는 2000년과 2020년 사이의 큰 차이는 없으며, median의 값이 2020년에 조금 오른 것을 확인할 수 있다.

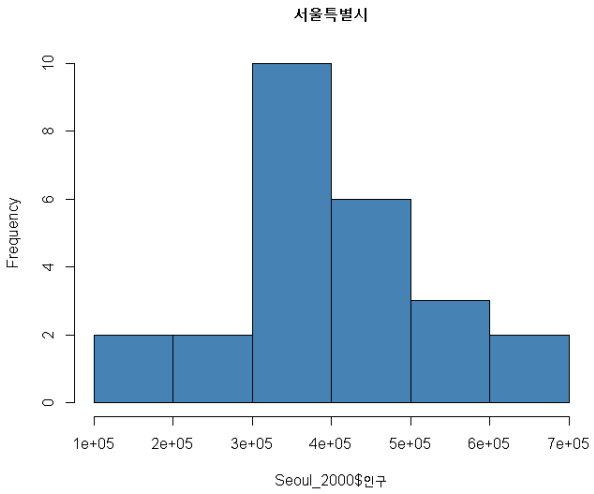
[울산]
울산의 경우 2000년과 비교하여 2020년에 수염이 사라지고 outlier가 2개 생겼으며 boxplot 상자의 길이가 매우 짧아졌음을 확인할 수 있다. 2000년에 울산의 최대, 최솟값은 각각 남구(323761명), 북구(118088명)이다. 2020년에 북구의 인구가 217051명으로 크게 증가했고 울주군 또한 160359명에서 225050명으로 눈에 띄게 증가했다. 이로 인해 2020년 인구수 데이터의 전체적 분포가 20만명 대를 중심으로 몰리게 되었고 이로 인해 outlier 값도 생겨났음을 유추할 수 있다.

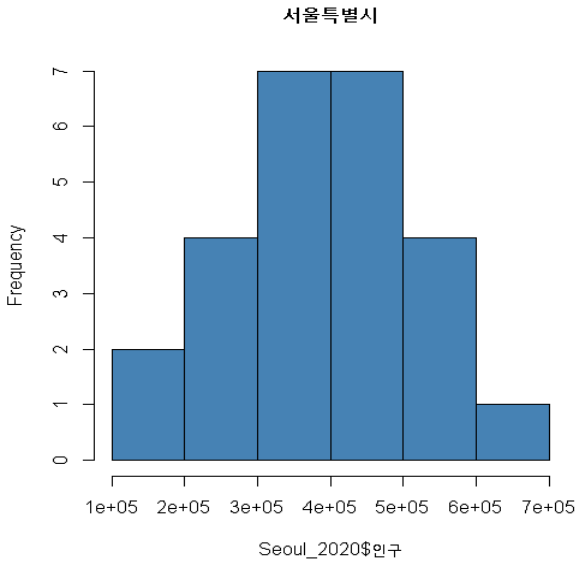
[인천]
인천의 경우 전체 인구수가 2000년(2466338명)에서 2020년(2945454명)으로 크게 증가했음을 알 수 있다. 인천은 2000년과 2020년 모두 outlier는 관측되지 않고 있으며 다른 광역시들과 비교

했을 때 box의 길이가 두드러지게 길다는 것을 알 수 있다. 옹진군, 강화군, 중구, 동구와 다른 인천의 구들 사이의 인구수 차이가 크기 때문에 boxplot의 길이가 길다는 것을 알 수 있다. 인천은 median 값이 2000년에 비해 2020년에 증가했고 box의 수염이 위쪽으로 짧아졌음을 확인할 수 있다.

다음으로, 각 연도와 시별로 아래와 같이 코드를 실행하여 skewness, kurtosis를 살펴보고 이를 비교해보았다.

[서울]





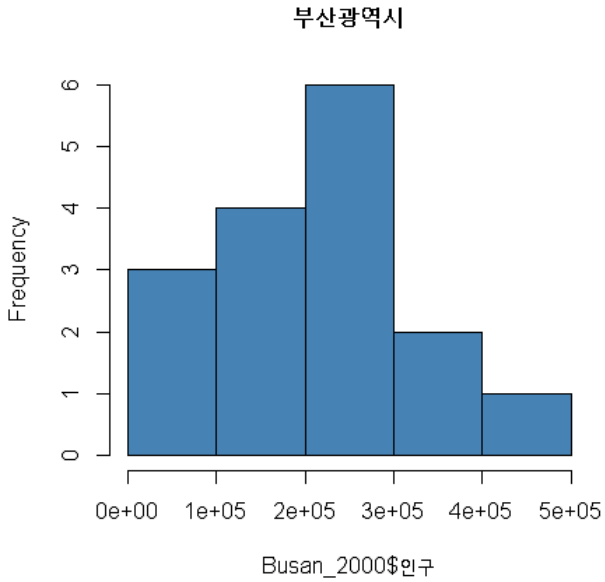

```
> library(moments)
> Seoul_2000 = pop2000[pop2000$지역명=='서울',]
> hist(Seoul_2000$인구, col='steelblue',
+      main='서울특별시')
> skewness(Seoul_2000$인구)
[1] -0.2093878
> kurtosis(Seoul_2000$인구)
[1] 3.106891
```

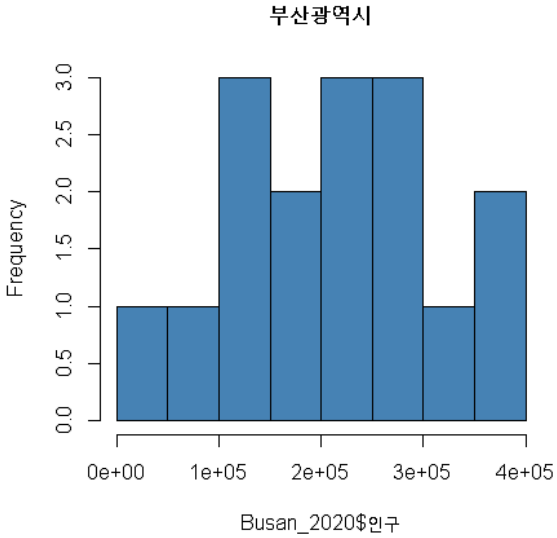
```
> Seoul_2020 = pop2020[pop2020$지역명=='서울특별시',]
> hist(Seoul_2020$인구, col='steelblue',
+      main='서울특별시')
> skewness(Seoul_2020$인구)
[1] -0.1628444
> kurtosis(Seoul_2020$인구)
[1] 2.912997
```

서울의 경우 skewness 값은 2000년(-0.21), 2020년(-0.16)정도로 모두 음(-)으로서 데이터의 중심이 정규분포보다 오른쪽으로 치우쳐져 있음을 알 수 있다.

kurtosis의 경우 2000년은 3.1 정도이고 2020년은 2.91 정도이다. 2000년에 분포의 뾰족한 형태가 2020년에 비해 조금 더 두드러지게 나타나고 있음을 확인할 수 있다.

[부산]



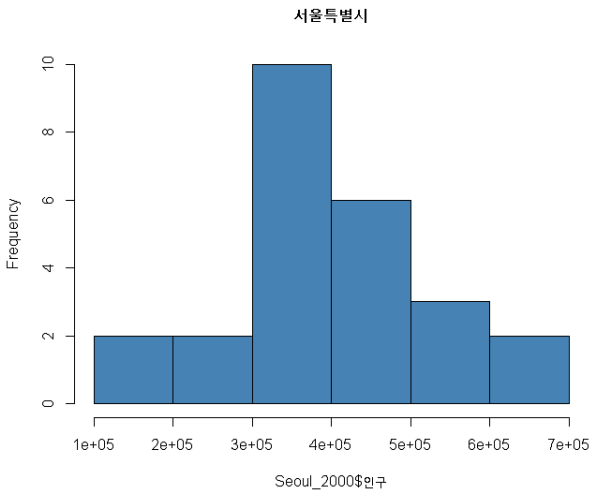


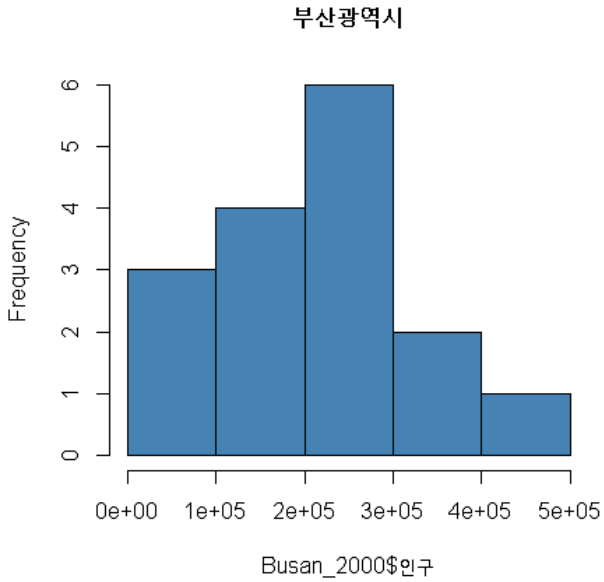

```
> Busan_2000 = pop2000[pop2000$지역명=='부산',]
> hist(Busan_2000$인구, col='steelblue',
+      main='부산광역시')
> skewness(Busan_2000$인구)
[1] -0.03979617
> kurtosis(Busan_2000$인구)
[1] 1.826998
> Busan_2020 = pop2020[pop2020$지역명=='부산광역시',]
> hist(Busan_2020$인구, col='steelblue',
+      main='부산광역시')
> skewness(Busan_2020$인구)
[1] 0.1060268
> kurtosis(Busan_2020$인구)
[1] 2.126159
```

부산의 경우 skewness 값은 2000년(-0.04), 2020년(+0.10) 정도임을 알 수 있다. 음(-)에서 양(+)으로 이동했음을 알 수 있다. 즉, Left Skewed 형태에서 Right Skewed의 형태로 변화했음을 알 수 있다.

kurtosis의 경우 2000년은 1.8 정도이고 2020년은 2.1 정도이다. 2020년에 분포의 뾰족한 형태가 2000년에 비해 조금 더 두드러지게 나타나고 있음을 확인할 수 있다. 즉, 2020년에 평균을 중심으로 더 가까이 몰려있음을 알 수 있다.

그렇다면 동일 연도 기준으로 서울과 부산의 차이를 살펴보자.

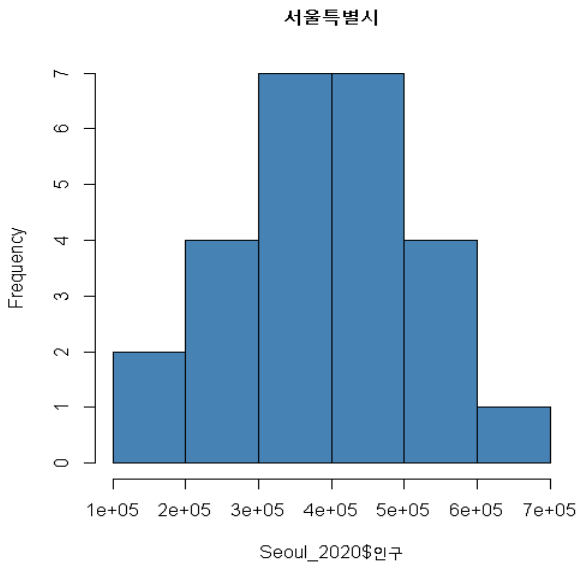


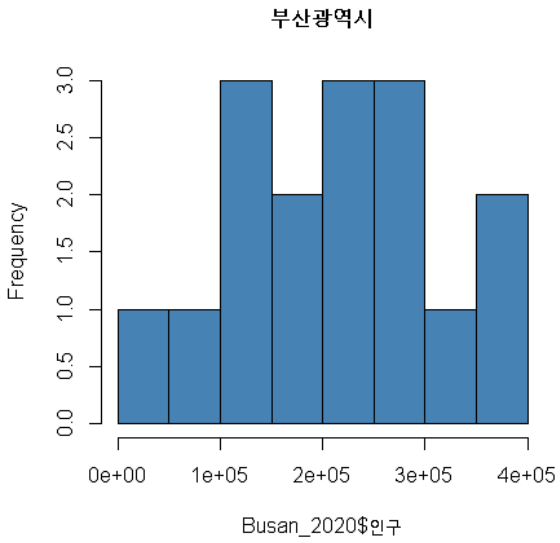



```
> library(moments)
> Seoul_2000 = pop2000[pop2000$지역명=='서울',]
> hist(Seoul_2000$인구, col='steelblue',
+      main='서울특별시')
> skewness(Seoul_2000$인구)
[1] -0.2093878
> kurtosis(Seoul_2000$인구)
[1] 3.106891
```

```
> Busan_2000 = pop2000[pop2000$지역명=='부산',]
> hist(Busan_2000$인구, col='steelblue',
+      main='부산광역시')
> skewness(Busan_2000$인구)
[1] -0.03979617
> kurtosis(Busan_2000$인구)
[1] 1.826998
```

2000년 서울의 경우 왜도 값은 -0.2이고 부산의 경우 -0.03으로 거의 0에 가까웠다.
kurtosis를 비교한 결과 서울의 첨도가 3.1, 부산이 1.8 정도로 서울의 첨도가 더 크다는 것을 알 수 있었다.





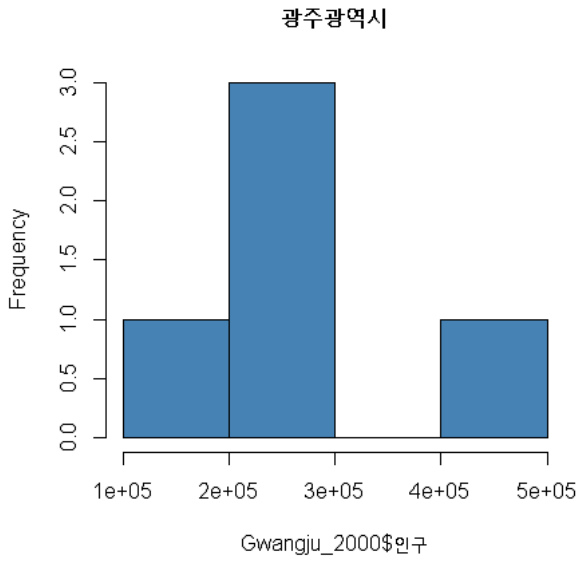
```
> Seoul_2020 = pop2020[pop2020$지역명=='서울특별시',]
> hist(Seoul_2020$인구, col='steelblue',
+      main='서울특별시')
> skewness(Seoul_2020$인구)
[1] -0.1628444
> kurtosis(Seoul_2020$인구)
[1] 2.912997
```

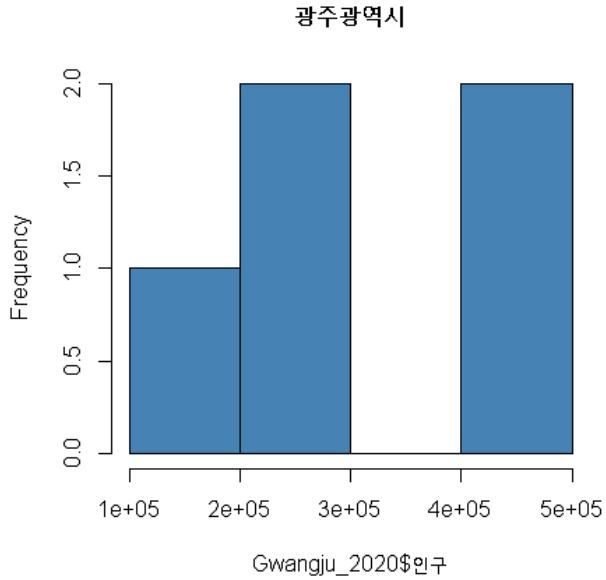
```
> Busan_2020 = pop2020[pop2020$지역명=='부산광역시',]
> hist(Busan_2020$인구, col='steelblue',
+      main='부산광역시')
> skewness(Busan_2020$인구)
[1] 0.1060268
> kurtosis(Busan_2020$인구)
[1] 2.126159
```

2020년 서울의 경우 왜도 값은 -0.16이고 부산의 경우 0.1이다. 즉, 2020년 부산의 경우 서울과 다르게 Right Skewed한 형태를 띠고 있음을 알 수 있다.
kurtosis를 비교한 결과 서울의 첨도가 2.9, 부산이 2.1 정도로 서울의 첨도가 조금 더 크다는 것을 알 수 있었다.

추가적으로, 서울/부산을 제외한 다른 시도 살펴보았다.

[광주]



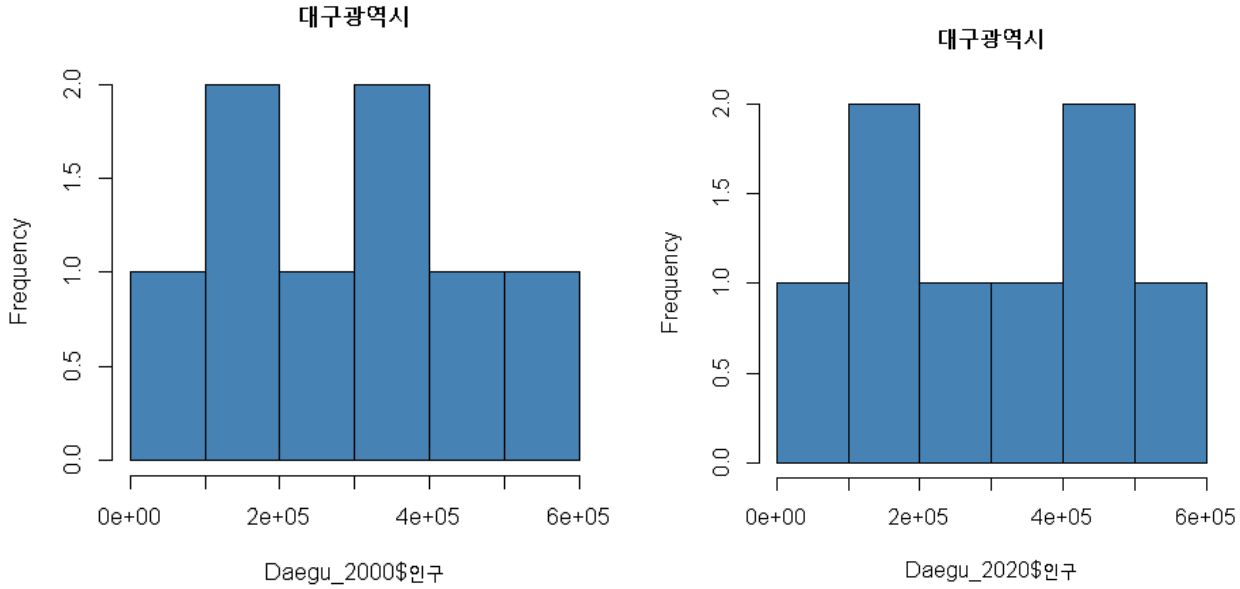



```
> Gwangju_2000 = pop2000[pop2000$지역명=='광주',]
> hist(Gwangju_2000$인구, col='steelblue',
+      main='광주광역시')
> skewness(Gwangju_2000$인구)
[1] 0.6414371
> kurtosis(Gwangju_2000$인구)
[1] 2.534638
> Gwangju_2020 = pop2020[pop2020$지역명=='광주광역시',]
> hist(Gwangju_2020$인구, col='steelblue',
+      main='광주광역시')
> skewness(Gwangju_2020$인구)
[1] -0.2312968
> kurtosis(Gwangju_2020$인구)
[1] 1.608932
```

광주의 경우 skewness 값은 2000년(0.64)에서 2020년(-0.23) 정도로 왜도의 부호가 양(+)에서

음(-)으로 이동했음을 알 수 있다.
kurtosis의 경우 2000년은 2.5 정도이고 2020년은 1.6 정도이다. 2000년에 분포의 뾰족한 형태가 2020년에 비해 더 두드러지게 나타나고 있음을 확인할 수 있다.

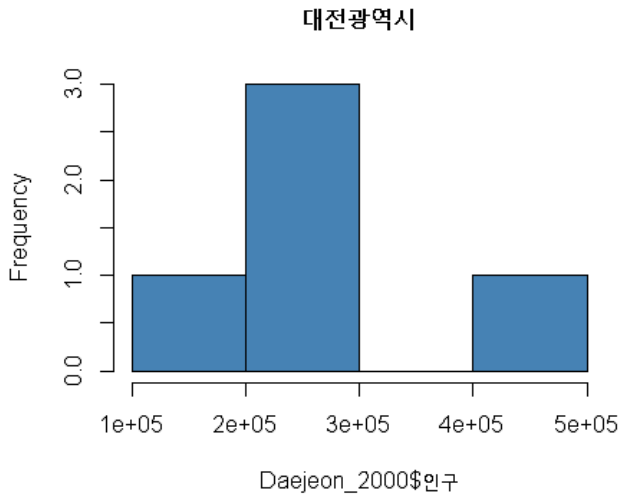
[대구]

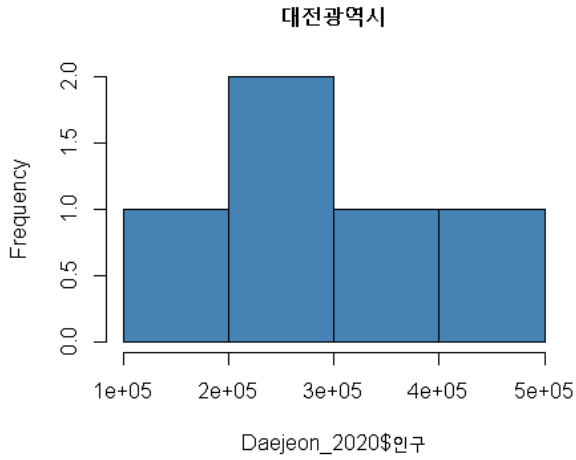



```
> Daegu_2000 = pop2000[pop2000$지역명=='대구',]
> hist(Daegu_2000$인구, col='steelblue',
+      main='대구광역시')
> skewness(Daegu_2000$인구)
[1] 0.2407861
> kurtosis(Daegu_2000$인구)
[1] 2.017732
> Daegu_2020 = pop2020[pop2020$지역명=='대구광역시',]
> hist(Daegu_2020$인구, col='steelblue',
+      main='대구광역시')
> skewness(Daegu_2020$인구)
[1] 0.1705404
> kurtosis(Daegu_2020$인구)
[1] 1.828938
```

대구의 경우 skewness 값은 2000년(0.24)에서 2020년(0.17) 정도로 모두 양(+)으로서 데이터의 중심이 정규분포보다 왼쪽으로 치우쳐져 있음을 알 수 있다. 즉, Right Skewed이며 2000년에서 조금 더 뚜렷하게 Right Skewed의 특징을 볼 수 있다.
kurtosis의 경우 2000년은 2.0 정도이고 2020년은 1.8 정도이다. 2000년에 분포의 뾰족한 형태가 2020년에 비해 조금 더 두드러지게 나타나고 있음을 확인할 수 있다.

[대전]



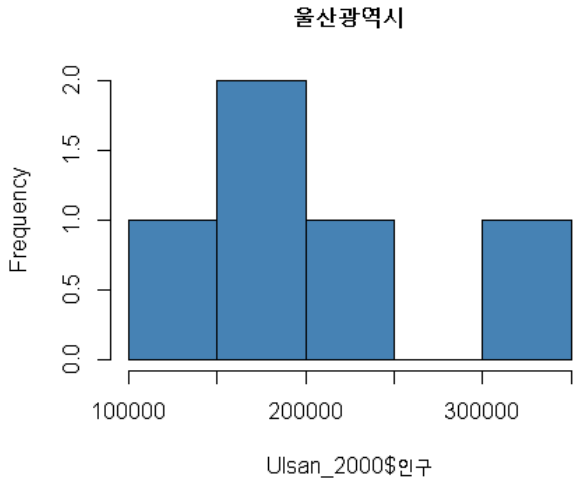


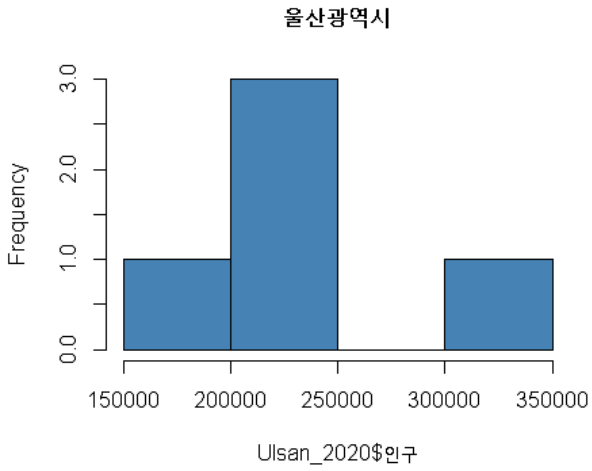

```
> Daejeon_2000 = pop2000[pop2000$지역명=='대전',]
> hist(Daejeon_2000$인구, col='steelblue',
+      main='대전광역시')
> skewness(Daejeon_2000$인구)
[1] 1.102431
> kurtosis(Daejeon_2000$인구)
[1] 2.863801
> Daejeon_2020 = pop2020[pop2020$지역명=='대전광역시',]
> hist(Daejeon_2020$인구, col='steelblue',
+      main='대전광역시')
> skewness(Daejeon_2020$인구)
[1] 0.5836154
> kurtosis(Daejeon_2020$인구)
[1] 1.809975
```

대전의 경우 skewness 값은 2000년(1.1)에서 2020년(0.58) 정도로 모두 양(+)으로서 데이터의 중심이 정규분포보다 왼쪽으로 치우쳐져 있음을 알 수 있다. 즉, Right Skewed이며 2000년에서 더 뚜렷하게 Right Skewed의 특징을 볼 수 있다.

kurtosis의 경우 2000년은 2.86 정도이고 2020년은 1.8 정도이다. 2000년에 분포의 뾰족한 형태가 2020년에 비해 더 뚜렷하게 나타나고 있음을 확인할 수 있다.

[울산]

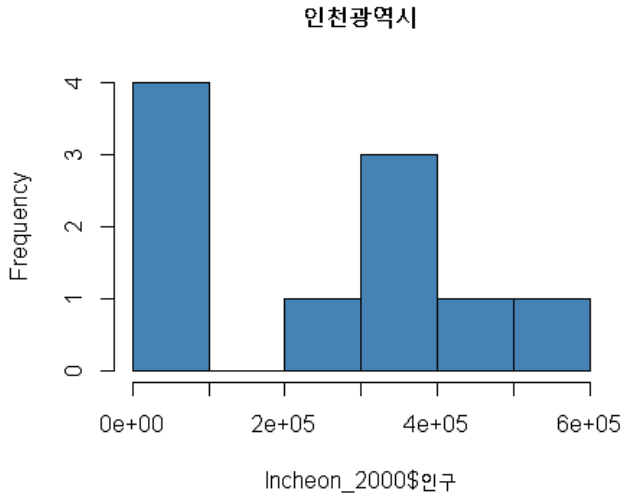


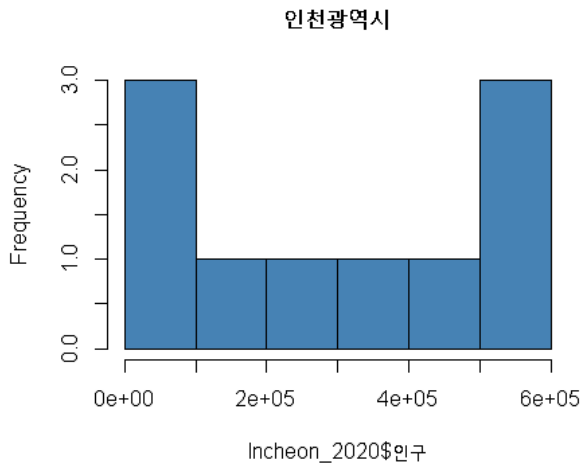



```
> Ulsan_2000 = pop2000[pop2000$지역명=='울산',]
> hist(Ulsan_2000$인구, col='steelblue',
+      main='울산광역시')
> skewness(Ulsan_2000$인구)
[1] 0.64854
> kurtosis(Ulsan_2000$인구)
[1] 2.237145
> Ulsan_2020 = pop2020[pop2020$지역명=='울산광역시',]
> hist(Ulsan_2020$인구, col='steelblue',
+      main='울산광역시')
> skewness(Ulsan_2020$인구)
[1] 0.6899044
> kurtosis(Ulsan_2020$인구)
[1] 2.622185
```

울산의 경우 2000년의 왜도는 0.64854, 2020년의 왜도는 0.6899로 거의 유사함을 알 수 있다. 또한, 첨도의 경우 2000년(2.237), 2020년(2.622)정도로 첨도 또한 거의 비슷했다.

[인천]





```
> Incheon_2000 = pop2000[pop2000$지역명=='인천',]
> hist(Incheon_2000$인구, col='steelblue',
+      main='인천광역시')
> skewness(Incheon_2000$인구)
[1] 0.01564393
> kurtosis(Incheon_2000$인구)
[1] 1.586081
> Incheon_2020 = pop2020[pop2020$지역명=='인천광역시',]
> hist(Incheon_2020$인구, col='steelblue',
+      main='인천광역시')
> skewness(Incheon_2020$인구)
[1] -0.1353928
> kurtosis(Incheon_2020$인구)
[1] 1.390927
```

인천의 경우 2000년의 왜도는 0.01로 거의 0에 가까우며 2020년의 왜도는 -0.1 정도였다.
첨도의 경우 2000년은 1.586, 2020년은 1.39 정도로 두 개 연도의 첨도가 유사함을 알 수 있다.

2. (R datasets:: ToothGrowth) 이의 성장에 비타민 C의 영향이 있는지 상자그림으로 분석하여라. notch사용.

본격적인 분석을 수행하기 전,?ToothGrowth를 실행하여 해당 데이터셋의 특징을 살펴보았다.

The Effect of Vitamin C on Tooth Growth in Guinea Pigs

Description

The response is the length of odontoblasts (cells responsible for tooth growth) in 60 guinea pigs. Each animal received one of three dose levels of vitamin C (0.5, 1, and 2 mg/day) by one of two delivery methods, orange juice or ascorbic acid (a form of vitamin C and coded as VC).

Usage

```
ToothGrowth
```

Format

A data frame with 60 observations on 3 variables.

[,1] len numeric Tooth length
[,2] supp factor Supplement type (VC or OJ).
[,3] dose numeric Dose in milligrams/day

Source

C. I. Bliss (1952). *The Statistics of Bioassay*. Academic Press.

References

McNeil, D. R. (1977). *Interactive Data Analysis*. New York: Wiley.

Crampton, E. W. (1947). The growth of the odontoblast of the incisor teeth as a criterion of vitamin C intake of the guinea pig. *The Journal of Nutrition*, **33**(5), 491–504. doi: 10.1093/jn/33.5.491.

첫 번째 column은 len으로서 tooth의 길이를, 두 번째 column은 supp factor로서 Guinea Pig에게 투여한 것이 VC(ascorbic acid)인지 OJ(오렌지 주스)인지를 나타내고 있다. 세 번째 column은 dose로서 투여량을 나타내고 있다.

```
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
> summary(ToothGrowth)
      len        supp
 Min.   : 4.20   OJ:30
 1st Qu.:13.07   VC:30
 Median :19.25
 Mean   :18.81
 3rd Qu.:25.27
 Max.   :33.90
      dose
 Min.   :0.500
 1st Qu.:0.500
 Median :1.000
 Mean   :1.167
 3rd Qu.:2.000
 Max.   :2.000
```

즉, 비타민C의 투여량(dose)과 방법(supp)에 따른 60마리 돼지들의 치아세포 길이 측정값을 나타내고 있는 데이터셋임을 알 수 있다. head()함수와 summary()함수를 통하여 데이터와 변수를 확인했다. 60마리 돼지 중 30마리에게는 OJ를 투여했고 나머지 30마리 돼지에게는 VC를 투여했음을 알 수 있다.

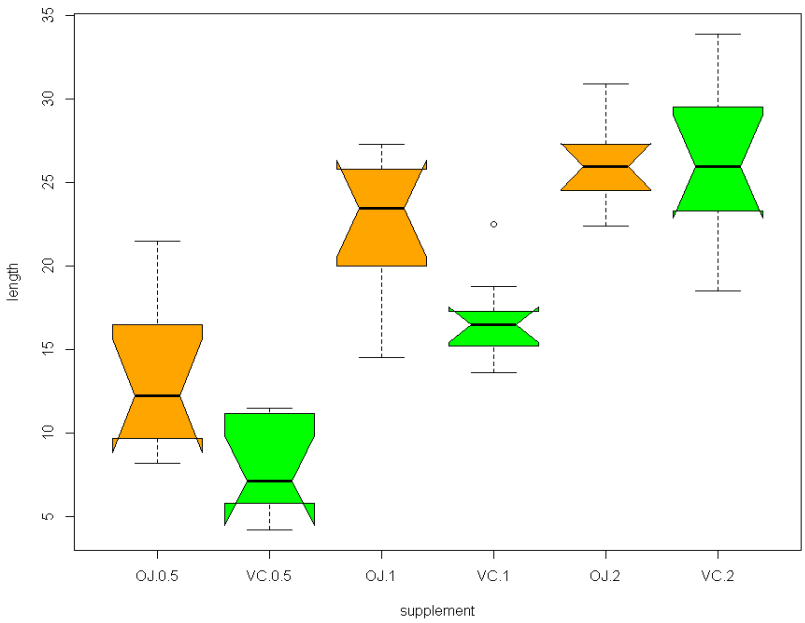
먼저, 투여량과 치아 길이 사이의 상관관계를 살펴보기 위해 아래의 코드를 실행했다.

```
> cor.test(ToothGrowth$len, ToothGrowth$dose)

        Pearson's product-moment correlation

data:  ToothGrowth$len and ToothGrowth$dose
t = 10.25, df = 58, p-value = 1.233e-14
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6892521 0.8777169
sample estimates:
      cor
0.8026913
```

해당 코드를 실행한 결과 피어슨 상관계수는 약 0.8 정도였다. p-value가 매우 작기 때문에 귀무가설을 기각할 수 있다. 결론적으로 복용량과 치아 성장 사이에 양의 상관관계가 있음을 알 수 있다.

다음으로, 아래의 코드를 통해 boxplot을 그려 해당 집단별로 len의 차이를 살펴보고자 했다.

```
boxplot(ToothGrowth$len~ToothGrowth$supp+ToothGrowth$dose,
        xlab="supplement",ylab="length",
        notch=TRUE,col=c("orange","green"))
```



boxplot을 그릴 때, notch라는 인자를 지정할 수 있는데 notch는 robust한 신뢰구간으로 중앙값에 대한 95% 신뢰구간을 계산하는 인자이다. 즉, 비교하는 대상의 notch의 범위가 겹치지 않는다면 대략 95%의 유의수준에서 두 집단은 유의미한 차이가 있다는 것을 알 수 있다.

주황색은 Orange juice, 초록색은 Ascorbic acid의 방법을 이용하여 투여한 데이터이다. x값의 OJ는 Orange juice, VC는 Ascorbic acid로서 옆의 숫자는 투여량을 의미한다. y값은 치아의 길이이다.

먼저, Ascorbic acid(초록색)를 살펴보면 투여량을 0.5, 1, 2로 늘릴수록 치아의 길이가 늘어남을 확인할 수 있었다. 또한, notch를 통해 median 값이 점점 늘어나고 있음을 알 수 있고 신뢰구간이 겹치지 않는다는 것을 확인할 수 있다. 이를 통해 각각의 집단(투여량 0.5,1,2)이 유의미하게 다르며 이를 통해 투여량이 치아의 길이에 영향을 미친다는 결론을 내릴 수 있다. 즉, Ascorbic acid를 통한 방법은 투여량이 늘어날수록 치아의 길이가 길어진다고 판단할 수 있다.

다음으로, Orange juice(주황색)를 살펴보면 notch를 통해 투여량이 0.5에서 1로 증가하였을 때는 신뢰구간이 겹치지 않지만 1에서 2로 증가하였을 때는 신뢰구간이 겹치는 것을 확인할 수 있다. 즉, Orange juice를 통한 방법은 투여량을 0.5에서 1로 늘렸을 때만(1에서 2로 늘렸을 때는 제외) 치아의 길이 성장에 유의미한 영향을 준다는 것을 알 수 있다.

다음으로, 같은 투여량으로 투여 방법 사이의 차이를 확인해보았다. 투여량이 0.5와 2인 경우에는 notch가 겹치고 있지만 1인 경우에는 notch가 겹치지 않는다. 즉, Orange juice 치아 길이

의 median이 Ascorbic acid에 비해 높다는 것을 알 수 있다. 결론적으로 투여량이 1인 경우에만 한정하여 투여 방법 사이의 유의미한 차이가 있음을 알 수 있다.

3. R에서 summary()와 fivenum()의 차이를 Nile 자료로 알아보아라.

Nile 데이터를 활용하여 summary(), fivenum() 그리고 quantile() 함수를 실행해본 결과는 아래와 같다.

```
> Nile
Time Series:
Start = 1871
End = 1970
Frequency = 1
  [1] 1120 1160  963 1210 1160 1160  813 1230 1370
 [10] 1140  995  935 1110  994 1020  960 1180  799
 [19]  958 1140 1100 1210 1150 1250 1260 1220 1030
 [28] 1100  774  840  874  694  940  833  701  916
 [37]  692 1020 1050  969  831  726  456  824  702
 [46] 1120 1100  832  764  821  768  845  864  862
 [55]  698  845  744  796 1040  759  781  865  845
 [64]  944  984  897  822 1010  771  676  649  846
 [73]  812  742  801 1040  860  874  848  890  744
 [82]  749  838 1050  918  986  797  923  975  815
 [91] 1020  906  901 1170  912  746  919  718  714
[100]  740
> summary(Nile)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  456.0   798.5   893.5   919.4  1032.5  1370.0
> fivenum(Nile)
[1]  456.0  798.0  893.5 1035.0 1370.0
> quantile(Nile)
    0%    25%    50%    75%   100%
 456.0  798.5  893.5 1032.5 1370.0
```

☑ summary()를 통해 평균값을 알 수 있지만 fivenum()에서는 알 수 없다.

summary()를 통해서 Nile 데이터셋의 Mean은 919.4임을 알 수 있지만 fivenum()을 통해서는 알 수 없었다.

☑ 제1사분위수(25%)와 제3사분위수(75%)의 값이 다르다(계산 방법의 차이)

summary()의 제1사분위수와 제3사분위수는 quantile()을 실행했을 때와 동일하다. 그러나 fivenum()을 실행했을 때는 798.0 그리고 1035.0으로 조금 다르게 산출되었음을 확인할 수 있었다.

fivenum()과 summary() 각각의 제1사분위수, 제3사분위수 산출 방법을 살펴본 결과는 다음과 같다.

- fivenum()

1) 자료 오름차순 정렬

2) 중앙값을 찾는다.

자료의 크기가 홀수(n)라면 중앙값은 (n+1)/2번째 값

자료의 크기가 짝수(n)라면 중앙값은 n/2번째 값과 (n+1)/2번째 값의 평균

3) 중앙값 아래의 데이터에 대하여 다시 중앙값을 찾는다. -> 제1분위수
중앙값 위의 데이터에 대하여 다시 중앙값을 찾는다. -> 제3분위수

- summary()
1) 자료를 오름차순으로 정렬
2) q% quantile의 위치는 1+(n-1)*q/100
3) 이 위치를 j라고 하면 i = floor(j), k = ceiling(j) 일 때,
q% quantile은 i번째 값+(j-i)(k번째 값-i번째 값)

이해를 돕기 위해 [1,3,5,7] 4개(짝수)의 숫자로 이루어진 리스트 그리고 [1,4,6,8,11,14,17] 7개(홀수)의 숫자로 이루어진 리스트에 각각 fivenum() 그리고 summary() 함수를 적용해보았다.

```
> x = c(1,3,5,7)
> x
[1] 1 3 5 7
> fivenum(x)
[1] 1 2 4 6 7
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     2.5     4.0     4.0     5.5     7.0
>
```

c(1,3,5,7)의 경우 중앙값은 4이기 때문에 fivenum(x)을 적용해보면 중앙값 아래 1,3의 중앙값은 2이고 중앙값 위의 5,7의 중앙값은 6이기 때문에 fivenum(x) 결과는 1 2 4 6 7이 산출된다.
summary(x)의 경우 25% quantile의 위치는 1+(4-1)*0.25 = 1+0.75 = 1.75이고 i=floor(1.75)=1 그리고 k=ceiling(1.75)=2이므로 25% quantile은
1번째값(1)+(1.75-1)*(3-1)=1+0.75*2 = 2.5가 Q1이 된다.
75% quantile의 위치는 1+(4-1)*0.75 = 1+ 3*0.75 = 1 + 2.25 = 3.25이고 i=floor(3.25)=3 그리고 k=ceiling(3.25) = 4이므로 75% quantile은
3번째값(1)+(3.25-3)*(7-5)=5+0.25*2 = 5.5가 Q3가 된다.

```
> y <- c(1,4,6,8,11,14,17)
> y
[1]  1  4  6  8 11 14 17
> fivenum(y)
[1]  1.0  5.0  8.0 12.5 17.0
> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.000   8.000   8.714  12.500  17.000
```

자료의 개수가 7개인 경우 (즉, 홀수인 경우) fivenum()과 summary()의 결과는 똑같이 산출되었음을 확인할 수 있다.

따라서, fivenum()과 summary()는 제1사분위수 그리고 제3사분위수 (quantile) 산출 방법이 다르다는 것을 통해 Nile 데이터셋의 경우 length가 100이기 때문에 짝수이므로 fivenum()과 summary()의 결과가 다르게 나옴을 확인했다. 만약, length가 101과 같이 홀수였다면 fivenum()과 summary()의 제 1사분위수 그리고 제3사분위수가 똑같이 산출되었을 것이다.

4. ~~(남겨두었던 숙제)~~

Ashwan에서 측정한 Nile 강의 유량 자료 (R datasets:: Nile)

문자전시를 만들고 결과를 설명하라. R에 함수가 없으니 원자료와 upward, downward rank를 구하여 세 개 열로 출력하여 depth에 맞는 값을 찾아 표를 수작업 또는 편집하여 완성하여라. (함수 스크립트를 짜면 시간이 많이 걸릴 수 있으니 의욕이 넘치는 학생은 일단 손으로 하고 '시간이 남으면' 도전하라. 0.5점 추가 점수.)

M은 median, H는 hinges, E는 eighths, D는 sixteenths를 의미한다.

EDA4 수업자료 4페이지를 통해 우리는 depth of median, hinges, eighths ...을 구할 수 있다. Nile 데이터셋의 경우 100개의 관측치로 구성된 자료임을 확인한 바 있다.

그렇다면 d(M)은 (1+100)/2=50.5이고 d(H)는 ([50.5]+1)/2=25.5임을 알 수 있다. 같은 원리로 d(E)=13, d(D)=7 d(C)는 4, d(B)는 2.5, d(A)는 1.5 그리고 마지막 문자값의 depth는 1로서 총 8개의 문자값이 필요함을알 수 있다. (M, H, E, D, C, B, A, 1)

이에 기반하여 R를 활용하여 함수 스크립트를 생성하고 Nile 데이터셋을 적용해보았다.

```r
func_4 <- function (df)
{
  letters <- c("M", "H", "E", "D", "C", "B", "A", "1")
  df <- sort(df)
  n <- length(df)

  downward_rnk <- vector()
  M_rnk <- vector()
  upward_rnk <- vector()

  downward_rnk[1] <- n
  M_rnk[1] <- n
  upward_rnk[1] <- n

  i = 1
  while(i <= 8){
    i=i + 1
    downward_rnk[i] <- floor(downward_rnk[i-1] + 1 ) /2 #floor : 소수점 버림
    M_rnk[i] <- floor(downward_rnk[i])
    upward_rnk[i] <- floor(downward_rnk[i] + 0.5)
  }

  lower <- (df[M_rnk[-1]] + df[upward_rnk[-1]])  / 2
  upper <- (df[n-M_rnk[-1] + 1] + df[n-upward_rnk[-1]+1]) / 2
  mid <- (lower + upper) / 2
  spread <- upper - lower

  #result 데이터프레임 생성
  result <- data.frame(letter=letters, depth=downward_rnk[-1],
                       lower, mid, upper, spread)
  return(result) #result 반환
}

###Nile 데이터셋 문자값 전시 result df 출력
func_4(Nile)
```

출력된 결과는 아래와 같다.

```
> ###Nile 데이터셋 문자값 전시 result df 출력
> func_4(Nile)
  letter depth lower    mid  upper spread
1      M  50.5 893.5 893.50  893.5    0.0
2      H  25.5 798.0 916.50 1035.0  237.0
3      E  13.0 742.0 946.00 1150.0  408.0
4      D   7.0 701.0 955.50 1210.0  509.0
5      C   4.0 692.0 961.00 1230.0  538.0
6      B   2.5 662.5 958.75 1255.0  592.5
7      A   1.5 552.5 933.75 1315.0  762.5
8      1   1.0 456.0 913.00 1370.0  914.0
> |
```

함수는 func_4로 지정했고 function()을 통해 생성했다. letters는 위에서 언급한대로 8개로 지정해주었고 sort를 통해 정렬 후 length를 n 변수에 저장해주었다. downward_rnk, M_rnk 그리고 upward_rnk는 vector를 통해 빈 벡터를 생성해주었다. while 문을 통해 연산을 수행했는데 문자값이 총 8개 필요했기 때문에 i<=8로 반복문을 실행시킬 횟수를 지정해주었다. 그리고 반복문을 수행하며 downward_rnk, M_rnk 그리고 upward_rnk의 벡터를 채워주었다. 다음으로 lower, upper, mid 그리고 spread 연산을 수행한 후 data.frame()을 통해 result 데이터프레임을 생성해주었고 return을 통해 반환해주었다. Nile데이터셋을 해당 함수에 적용해본 결과 result df이 올바르게 출력되었음을 확인했다.