Tests of Normality

coop711

2015년 3월 9일

BMI data revisited

- rn96 자료 읽어들이기

```
rn96<-read.table("rn96.txt",header=T,sep="")
head(rn96)
```

```
##   height weight
## 1    161     50
## 2    155     49
## 3    158     42
## 4    170     65
## 5    160     60
## 6    156     52
```

```
BMI<-round(rn96$weight/(rn96$height/100)^2,digits=1)
head(BMI)
```

```
## [1] 19.3 20.4 16.8 22.5 23.4 21.4
```

```
head(cbind(rn96,BMI))
```

```
##   height weight  BMI
## 1    161     50 19.3
## 2    155     49 20.4
## 3    158     42 16.8
## 4    170     65 22.5
## 5    160     60 23.4
## 6    156     52 21.4
```

```
head(data.frame(rn96,BMI))
```

```
##   height weight  BMI
## 1    161     50 19.3
## 2    155     49 20.4
## 3    158     42 16.8
## 4    170     65 22.5
## 5    160     60 23.4
## 6    156     52 21.4
```

- 정규분포 여부를 검증하는 nortest 패키지 설치

```
install.packages("nortest", repos="http://cran.nexr.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/_h/tg1th9bd4h98rjjb5vy9gn3m0000gn/T//RtmpfPTECh/downloaded_pac
kages
```

```
library(nortest)
```

```
## Warning: package 'nortest' was built under R version 3.1.3
```

- nortest 패키지의 설명문서 열어보기

```
help(package=nortest)
```

- 5가지 종류의 정규성 검증을 apply()함수를 이용하여 3개의 변수에 한꺼번에 적용

```
apply(cbind(rn96,BMI),2,ad.test)
```

```
## $height
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 0.28, p-value = 0.6267
##
##
## $weight
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 0.4278, p-value = 0.2978
##
##
## $BMI
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 0.518, p-value = 0.1778
```

```
apply(cbind(rn96,BMI),2,cvm.test)
```

```
## $height
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 0.0433, p-value = 0.6102
##
##
## $weight
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 0.0735, p-value = 0.2449
##
##
## $BMI
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 0.0803, p-value = 0.1994
```

```
apply(cbind(rn96,BMI),2,lillie.test)
```

```
## $height
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.0904, p-value = 0.5439
##
##
## $weight
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.1359, p-value = 0.05487
##
##
## $BMI
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.1195, p-value = 0.1485
```

```
apply(cbind(rn96,BMI),2,pearson.test)
```

```
## $height
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 9.2683, p-value = 0.159
##
##
## $weight
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 4.878, p-value = 0.5595
##
##
## $BMI
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 6.6341, p-value = 0.356
```

```
apply(cbind(rn96,BMI),2,sf.test)
```

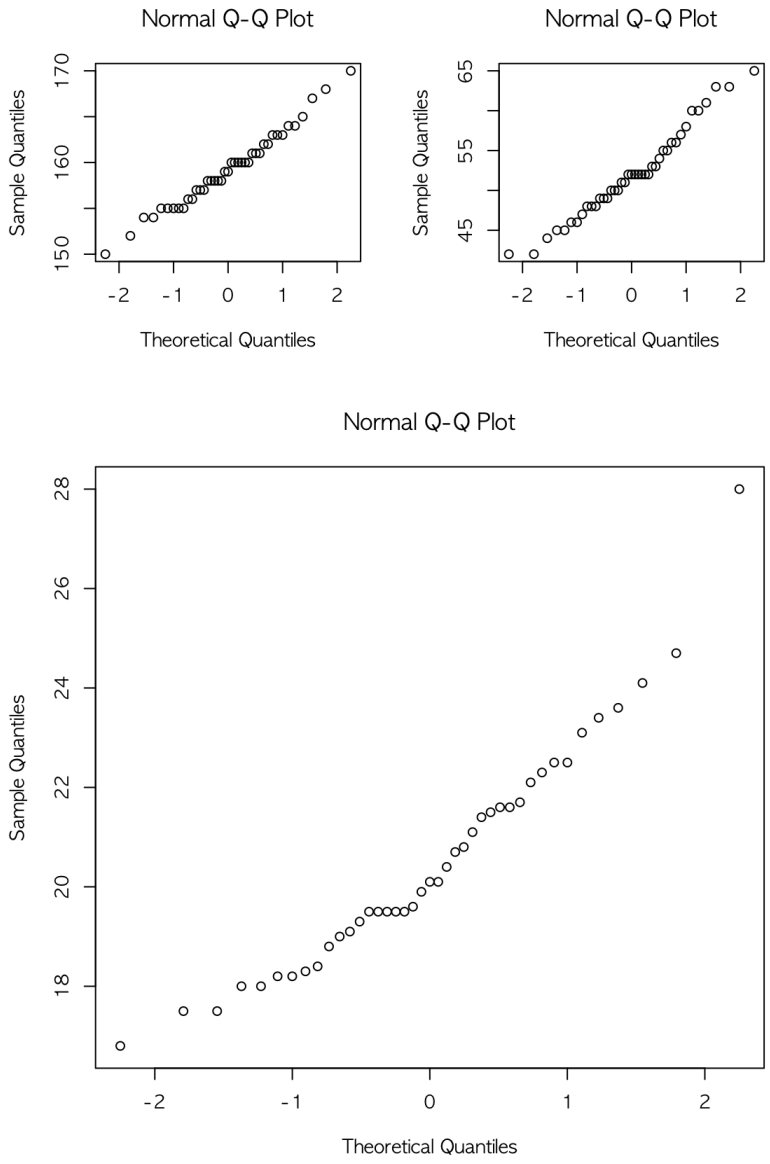
```
## $height
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9818, p-value = 0.6519
##
##
## $weight
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9744, p-value = 0.4011
##
##
## $BMI
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9424, p-value = 0.03893
```

```
apply(cbind(rn96,BMI),2,shapiro.test)
```

```
## $height
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9836, p-value = 0.8096
##
##
## $weight
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9694, p-value = 0.3304
##
##
## $BMI
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.9466, p-value = 0.05333
```

- apply() 를 qqnorm() 에도 적용할 수 있음. 단, 판을 미리 짜 놓아야 함.

```
layout(matrix(c(1,2,3,3),2,2, byrow=TRUE), heights=c(1,2))
apply(cbind(rn96,BMI),2,qqnorm)
```



```
## $height
## $height$x
##  [1]  0.44193453 -1.22782626 -0.37546177  2.25092570  0.06117541
##  [6] -0.73323578  0.65542351 -0.31060943 -0.24703899  1.54663527
## [11]  0.12258084 -1.10700288 -1.54663527 -1.00049055 -0.58139301
## [16] -0.51042164  0.18445244 -0.18445244  0.24703899  0.31060943
## [21] -1.79176429 -1.36985648 -2.25092570  0.51042164  0.73323578
## [26]  1.10700288  0.58139301 -0.90426734 -0.06117541  0.81576571
## [31]  0.00000000  0.37546177 -0.12258084  1.36985648 -0.65542351
## [36]  0.90426734 -0.81576571  1.22782626  1.00049055  1.79176429
## [41] -0.44193453
##
## $height$y
##  [1] 161 155 158 170 160 156 162 158 158 167 160 155 154 155 157 157 160
## [18] 158 160 160 152 154 150 161 162 164 161 155 159 163 159 160 158 165
## [35] 156 163 155 164 163 168 157
##
##
## $weight
## $weight$x
##  [1] -0.37546177 -0.58139301 -2.25092570  2.25092570  1.10700288
##  [6] -0.06117541  1.00049055 -1.10700288 -1.36985648 -0.18445244
## [11] -0.31060943 -1.79176429  0.37546177  0.00000000 -0.81576571
## [16] -0.73323578 -0.51042164  0.06117541 -0.12258084  0.44193453
## [21] -1.54663527  0.73323578  1.54663527  0.12258084  0.90426734
## [26] -0.44193453  0.18445244  0.51042164 -1.00049055 -0.24703899
## [31]  1.36985648  0.58139301 -1.22782626  1.79176429  1.22782626
## [36]  0.81576571  0.24703899 -0.90426734  0.31060943  0.65542351
## [41] -0.65542351
##
## $weight$y
##  [1] 50 49 42 65 60 52 58 46 45 51 50 42 53 52 48 48 49 52 51 53 44 56 63
## [24] 52 57 49 52 54 46 50 61 55 45 63 60 56 52 47 52 55 48
##
##
## $BMI
## $BMI$x
##  [1] -0.51042164  0.12258084 -2.25092570  0.90426734  1.22782626
##  [6]  0.37546177  0.73323578 -0.81576571 -1.36985648 -0.90426734
## [11] -0.44193453 -1.79176429  0.81576571  0.51042164 -0.37546177
## [16] -0.31060943 -0.58139301  0.24703899 -0.06117541  0.18445244
## [21] -0.65542351  1.36985648  2.25092570  0.00000000  0.65542351
## [26] -1.10700288  0.06117541  1.00049055 -1.00049055 -0.73323578
## [31]  1.54663527  0.44193453 -1.22782626  1.10700288  1.79176429
## [36]  0.31060943  0.58139301 -1.54663527 -0.12258084 -0.24703899
## [41] -0.18445244
##
## $BMI$y
##  [1] 19.3 20.4 16.8 22.5 23.4 21.4 22.1 18.4 18.0 18.3 19.5 17.5 22.3 21.6
## [15] 19.5 19.5 19.1 20.8 19.9 20.7 19.0 23.6 28.0 20.1 21.7 18.2 20.1 22.5
## [29] 18.2 18.8 24.1 21.5 18.0 23.1 24.7 21.1 21.6 17.5 19.6 19.5 19.5
```

Quetelet의 가슴둘레자료 정규분포 적합도

자료 구성

- Quetelet 의 원본 테이블로부터

```
chest<-33:48
freq<-c(3,18,81,185,420,749,1073,1079,934,658,370,92,50,21,4,1)
sum(freq)
```

```
## [1] 5738
```

- 5738명의 가슴둘레 자료임. 실제 측정치를 가까운 정수값으로 반올림하였음을 기억해 둘 필요.
- data frame 으로 재구성

```
quetelet.chest<-data.frame(chest,freq)
```

- 케틀레가 작업한 바와 같이 히스토그램 형태로 나타내기 위해서는 한 줄의 벡터로 변환하여야 함. for loop 를 이용하기 위하여 다음 작업 수행.

```
quetelet.chest.long<-rep(33,3)
for (i in 34:48) {
quetelet.chest.long<-c(quetelet.chest.long,rep(i,quetelet.chest$freq[i-32]))
}
length(quetelet.chest.long)
```

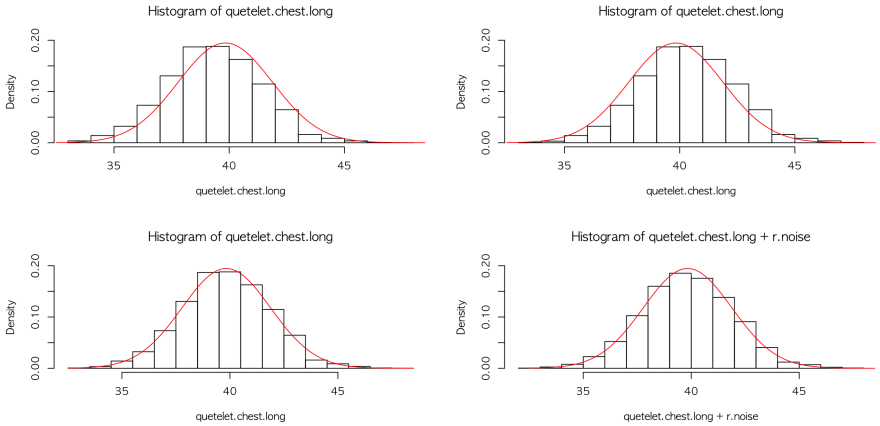
```
## [1] 5738
```

- 히스토그램과 정규분포 곡선과 비교하기 위하여 가슴둘레 자료의 평균과 표준편차를 계산한 뒤 밀도함수를 그리기 위한 좌표 마련

```
mean.chest<-mean(quetelet.chest.long)
sd.chest<-sd(quetelet.chest.long)
x<-seq(32.5,48.5,length=1000)
y.norm<-dnorm(x,mean=mean.chest,sd=sd.chest)
```

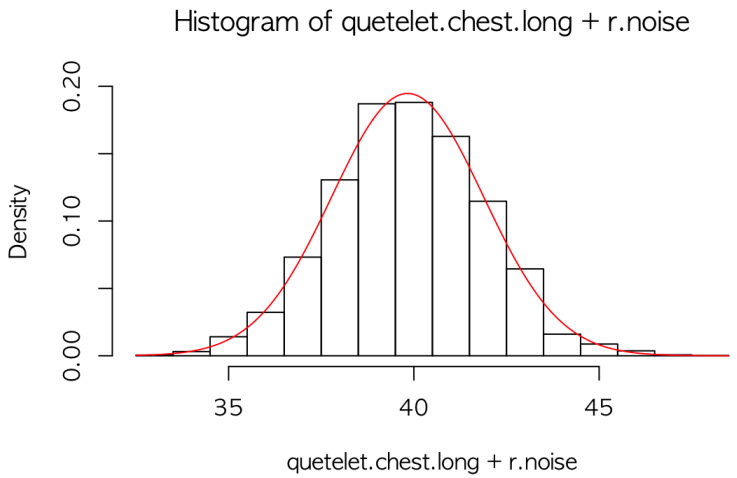
- 다음 네 장의 그림을 비교하면 어떤 것이 가장 자료의 특징을 잘 나타낸다고 볼 수 있는가? 함께 그린 정규곡선 밀도함수를 보고 판단하시오.

```
par(mfrow=c(2,2))
h1<-hist(quetelet.chest.long,prob=T,ylim=c(0,0.2))
lines(x,y.norm,col="red")
h2<-hist(quetelet.chest.long,prob=T,right=F,ylim=c(0,0.2))
lines(x,y.norm,col="red")
h3<-hist(quetelet.chest.long,prob=T,breaks=32.5:48.5,ylim=c(0,0.2))
lines(x,y.norm,col="red")
r.noise<-runif(5738)-0.5
h4<-hist(quetelet.chest.long+r.noise,prob=T,ylim=c(0,0.2))
lines(x,y.norm,col="red")
```



- 랜덤 노이즈를 더하고 breaks도 조정하면

```
par(mfrow=c(1,1))
h5<-hist(quetelet.chest.long+r.noise,prob=T,breaks=32.5:48.5,ylim=c(0,0.2))
lines(x,y.norm,col="red")
```



- 각각의 히스토그램들을 그릴 때 사용한 breaks와 counts 값을 추적

```
h1
```

```
## $breaks
##  [1] 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
##
## $counts
##  [1]   21   81  185  420  749 1073 1079  934  658  370   92   50   21    4
## [15]    1
##
## $density
##  [1] 0.0036598118 0.0141164169 0.0322411990 0.0731962356 0.1305332869
##  [6] 0.1869989543 0.1880446148 0.1627744859 0.1146741025 0.0644823980
## [11] 0.0160334611 0.0087138376 0.0036598118 0.0006971070 0.0001742768
##
## $mids
##  [1] 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5 45.5 46.5
## [15] 47.5
##
## $xname
## [1] "quetelet.chest.long"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
list(h1$breaks,h1$counts)
```

```
## [[1]]
##  [1] 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
##
## [[2]]
##  [1]   21   81  185  420  749 1073 1079  934  658  370   92   50   21    4
## [15]    1
```

```
list(h2$breaks,h2$counts)
```

```
## [[1]]
##  [1] 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
##
## [[2]]
##  [1]    3   18   81  185  420  749 1073 1079  934  658  370   92   50   21
## [15]    5
```

```
list(h3$breaks,h3$counts)
```

```
## [[1]]
##  [1] 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5 45.5
## [15] 46.5 47.5 48.5
##
## [[2]]
##  [1]    3   18   81  185  420  749 1073 1079  934  658  370   92   50   21
## [15]    4    1
```

```
list(h4$breaks,h4$counts)
```

```
## [[1]]
##  [1] 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
##
## [[2]]
##  [1]    2   14   45  130  299  589  918 1064 1007  794  521  232   69   40
## [15]   11    3
```

```
list(h5$breaks,h5$counts)
```

```
## [[1]]
##  [1] 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5 44.5 45.5
## [15] 46.5 47.5 48.5
##
## [[2]]
##  [1]    3   18   81  185  420  749 1073 1079  934  658  370   92   50   21
## [15]    4    1
```

- 정규분포 테스트를 적용해 보면?

```
quetelet.chest.noise<-quetelet.chest.long+r.noise
apply(cbind(quetelet.chest.long,quetelet.chest.noise),2,ad.test)
```

```
## $quetelet.chest.long
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 55.6932, p-value < 2.2e-16
##
##
## $quetelet.chest.noise
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 0.6184, p-value = 0.1075
```

```
apply(cbind(quetelet.chest.long,quetelet.chest.noise),2,cvm.test)
```

```
## Warning in FUN(newX[, i], ...): p-value is smaller than 7.37e-10, cannot
## be computed more accurately
```

```
## $quetelet.chest.long
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 10.5821, p-value = 7.37e-10
##
##
## $quetelet.chest.noise
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 0.0594, p-value = 0.385
```

```
apply(cbind(quetelet.chest.long,quetelet.chest.noise),2,lillie.test)
```

```
## $quetelet.chest.long
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.0983, p-value < 2.2e-16
##
##
## $quetelet.chest.noise
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.0093, p-value = 0.2616
```

```
apply(cbind(quetelet.chest.long,quetelet.chest.noise),2,pearson.test)
```

```
## $quetelet.chest.long
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 45056.67, p-value < 2.2e-16
##
##
## $quetelet.chest.noise
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 104.9613, p-value = 0.0004009
```

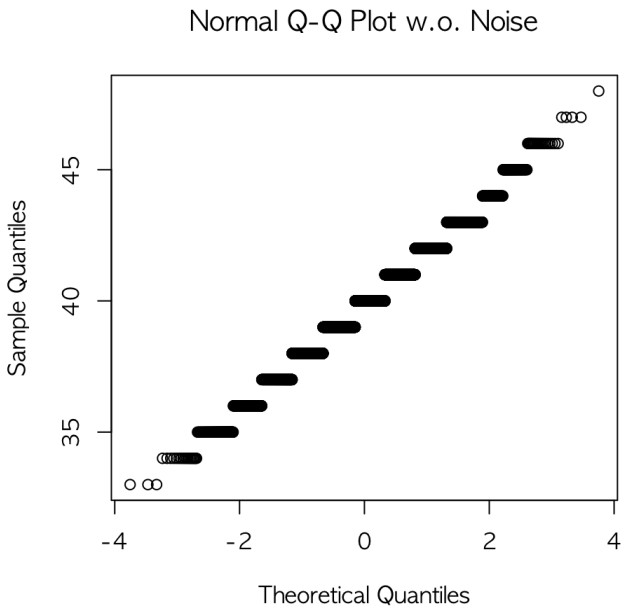
- sf.test()는 크기가 5000이하인 경우에만 사용할 수 있으므로 랜덤표본 추출 후 적용

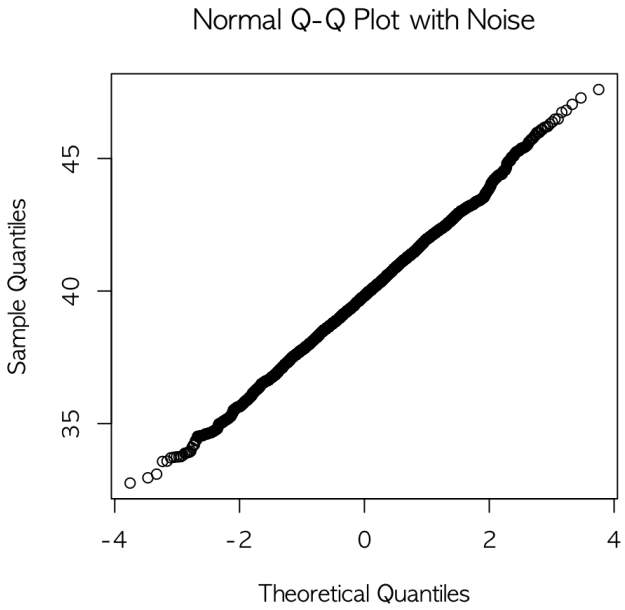
```
id.sample<-sample(1:5738,size=5000)
quetelet.chest.long.sample<-quetelet.chest.long[id.sample]
quetelet.chest.noise.sample<-quetelet.chest.noise[id.sample]
apply(cbind(quetelet.chest.long.sample,quetelet.chest.noise.sample),2,sf.test)
```

```
## $quetelet.chest.long.sample
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9795, p-value < 2.2e-16
##
##
## $quetelet.chest.noise.sample
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9992, p-value = 0.02219
```

- qqnorm() 을 그려보면

```
par(mfrow=c(2,1))
qqnorm(quetelet.chest.long, main="Normal Q-Q Plot w.o. Noise")
qqnorm(quetelet.chest.noise, main="Normal Q-Q Plot with Noise")
```


Normal Q-Q Plot w.o. Noise
Sample Quantiles
35
40
45
-4
-2
0
2
4
Theoretical Quantiles


Normal Q-Q Plot with Noise
Sample Quantiles
35
40
45
-4
-2
0
2
4
Theoretical Quantiles