Quetelet Chest Data : Tests of Normality

coop711

2018-04-03

Quetelet의 가슴둘레자료 정규분포 적합도

`nortest` 패키지 설치

```
# install.packages("nortest", repos="https://cran.rstudio.com/")
library(nortest)
```

nortest 패키지의 설명문서 열어보기

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

`ad.test`, `cvm.test`, `lillie.test` 등은 모두 EDF 기반의 도구임. 기본적으로 표본분포함수와 정규분포함수를 비교하는 것임.

Data

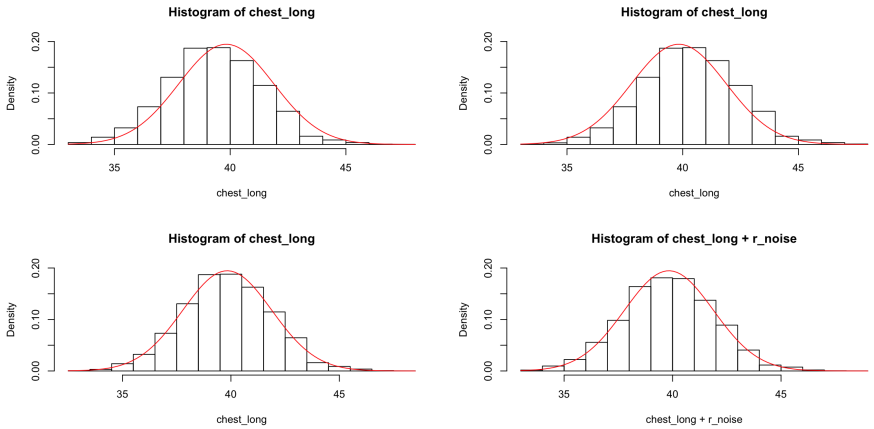
```
load("./Quetelet_chest.RData")
ls()
```

```
##  [1] "chest"       "chest_long"  "chest_table" "curve_df"    "freq"
##  [6] "g0"          "g1"          "g2"          "g3"          "g4"
## [11] "g5"          "g6"          "h_chest"     "h_chest.2"   "main_title"
## [16] "mean_chest"  "poly_df"     "sd_chest"    "sub_title"   "total"
## [21] "x_area"      "x_chest"     "x_coord"     "x_lab"       "x_lower"
## [26] "x_upper"     "y"           "y_coord"     "y_lab"       "y_norm"
```

Histogram

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

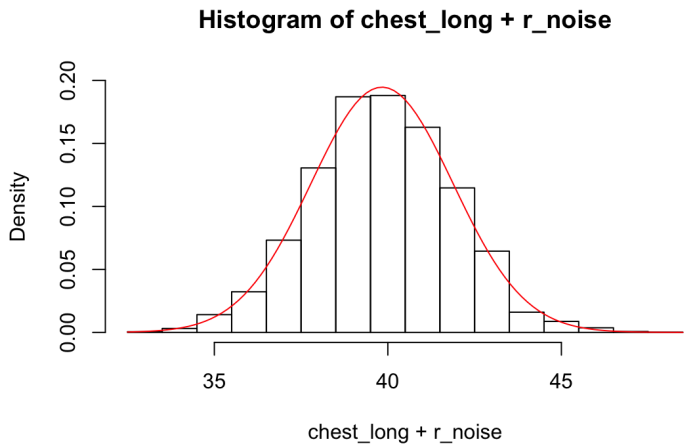
```
par(mfrow = c(2, 2))
x <- x_chest
h1 <- hist(chest_long,
           prob = TRUE,
           ylim = c(0, 0.2))
curve(dnorm(x, mean_chest, sd_chest),
      add = TRUE,
      col = "red")
h2 <- hist(chest_long,
           prob = TRUE,
           right = FALSE,
           ylim = c(0, 0.2))
curve(dnorm(x, mean_chest, sd_chest),
      add = TRUE,
      col = "red")
h3 <- hist(chest_long,
           prob = TRUE,
           breaks = 32.5:48.5,
           ylim = c(0, 0.2))
curve(dnorm(x, mean_chest, sd_chest),
      add = TRUE,
      col = "red")
r_noise <- runif(5738) - 0.5
h4 <- hist(chest_long + r_noise,
           prob = TRUE,
           ylim = c(0, 0.2))
curve(dnorm(x, mean_chest, sd_chest),
      add = TRUE,
      col = "red")
```



Random Noise

랜덤 노이즈를 더하고 `breaks` 도 조정하면

```
par(mfrow = c(1, 1))
h5 <- hist(chest_long + r_noise,
           prob = TRUE,
           breaks = 32.5:48.5,
           ylim = c(0, 0.2))
curve(dnorm(x, mean_chest, sd_chest),
      add = TRUE,
      col = "red")
```



`breaks` and `counts`

각각의 히스토그램들을 그릴 때 사용한 `breaks` 와 `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] "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] 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
##
## [[2]]
##  [1]   10   55  127  319  564  941 1038 1029  788  511  233   66   43   11
## [15]    2    1
```

```
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
```

nortest

정규분포 테스트를 적용해 보면?

```
chest_noise <- chest_long + r_noise
apply(cbind(chest_long, chest_noise),
      MARGIN = 2,
      FUN = ad.test)
```

```
## $chest_long
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 55.693, p-value < 2.2e-16
##
##
## $chest_noise
##
##  Anderson-Darling normality test
##
## data:  newX[, i]
## A = 0.66957, p-value = 0.08037
```

```
apply(cbind(chest_long, chest_noise),
      MARGIN = 2,
      FUN = cvm.test)
```

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

```
## $chest_long
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 10.582, p-value = 7.37e-10
##
##
## $chest_noise
##
##  Cramer-von Mises normality test
##
## data:  newX[, i]
## W = 0.076648, p-value = 0.2291
```

```
apply(cbind(chest_long, chest_noise),
      MARGIN = 2,
      FUN = lillie.test)
```

```
## $chest_long
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.098317, p-value < 2.2e-16
##
##
## $chest_noise
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  newX[, i]
## D = 0.011393, p-value = 0.08704
```

```
apply(cbind(chest_long, chest_noise),
      MARGIN = 2,
      FUN = pearson.test)
```

```
## $chest_long
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 45057, p-value < 2.2e-16
##
##
## $chest_noise
##
##  Pearson chi-square normality test
##
## data:  newX[, i]
## P = 124.9, p-value = 2.691e-06
```

sf.test

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

```
id_sample <- sample(1:5738, size = 5000)
chest_long_sample <- chest_long[id_sample]
chest_noise_sample <- chest_noise[id_sample]
apply(cbind(chest_long_sample, chest_noise_sample),
      MARGIN = 2,
      FUN = sf.test)
```

```
## $chest_long_sample
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.9794, p-value < 2.2e-16
##
##
## $chest_noise_sample
##
##  Shapiro-Francia normality test
##
## data:  newX[, i]
## W = 0.99925, p-value = 0.02375
```

qqnorm()

`qqnorm()` 을 그려보면

```
par(mfrow = c(1, 2))
qqnorm(chest_long,
       main = "Normal Q-Q Plot w.o. Noise")
qqnorm(chest_noise,
       main = "Normal Q-Q Plot with Noise")
```