Goodness of Fit Test

coop711

2015년 3월 14일

Matching Problems

- 매칭 문제에서 관찰한 값들을 data frame 으로 저장.

```
options(digits=3)
n.matching<-c(0,1,2,4)
o.matching<-c(22,12,15,1)
p.matching<-c(9,8,6,1)/24
matching<-data.frame(n.matching, o.matching, p.matching)
matching
```

```
##   n.matching o.matching p.matching
## 1          0         22     0.3750
## 2          1         12     0.3333
## 3          2         15     0.2500
## 4          4          1     0.0417
```

- 표본평균을 계산하여 $1 \pm 1/\sqrt{50}$ 에 들어가는 지 확인.

```
mean.matching<-sum(n.matching*o.matching/sum(o.matching))
mean.matching
```

```
## [1] 0.92
```

- p.matching 으로 계산한 확률모델에 부합하는지 적합도 검정 수행. 각종 검정통계량 확인. warning() 이 나온 이유가 무엇인지 함께 산출된 통계값들을 근거로 파악.

```
chisq.test.matching<-chisq.test(x=o.matching,p=p.matching)
```

```
## Warning in chisq.test(x = o.matching, p = p.matching): Chi-squared
## approximation may be incorrect
```

```
chisq.test.matching
```

```
##
##  Chi-squared test for given probabilities
##
## data:  o.matching
## X-squared = 2.93, df = 3, p-value = 0.402
```

```
chisq.test.matching$statistic
```

```
## X-squared
##      2.93
```

```
chisq.test.matching$parameter
```

```
## df
##  3
```

```
chisq.test.matching$p.value
```

```
## [1] 0.402
```

```
chisq.test.matching$method
```

```
## [1] "Chi-squared test for given probabilities"
```

```
chisq.test.matching$data.name
```

```
## [1] "o.matching"
```

```
chisq.test.matching$observed
```

```
## [1] 22 12 15  1
```

```
chisq.test.matching$expected
```

```
## [1] 18.75 16.67 12.50  2.08
```

```
chisq.test.matching$residuals
```

```
## [1]  0.751 -1.143  0.707 -0.751
```

```
chisq.test.matching$stdres
```

```
## [1]  0.949 -1.400  0.816 -0.767
```

- 검정통계량을 계산하고, p-value를 찾는 과정을 단계별로 살펴보자.

```
sum(o.matching)
```

```
## [1] 50
```

```
e.matching<-50*p.matching
e.matching
```

```
## [1] 18.75 16.67 12.50  2.08
```

```
(o.matching-e.matching)**2/e.matching
```

```
## [1] 0.563 1.307 0.500 0.563
```

```
sum((o.matching-e.matching)**2/e.matching)
```

```
## [1] 2.93
```

```
chisq.matching<-sum((o.matching-e.matching)**2/e.matching)
chisq.matching
```

```
## [1] 2.93
```

```
p.value<-1-pchisq(chisq.matching, df=3)
p.value
```

```
## [1] 0.402
```

Lottery Data의 Uniformity Test

- lottery 자료 읽어들이기. 기초통계 확인.

```
lottery<-read.table("lottery.txt",header=TRUE)
head(lottery)
```

```
##   lottery.number lottery.payoff
## 1            810            190
## 2            156            120
## 3            140            286
## 4            542            184
## 5            507            384
## 6            972            324
```

```
attach(lottery)
```

- lottery의 시행과정을 듣고 각 기초통계 값이 이론적으로 기대하는 값과 잘 들어맞는지 판단.

```
summary(lottery.number)
```

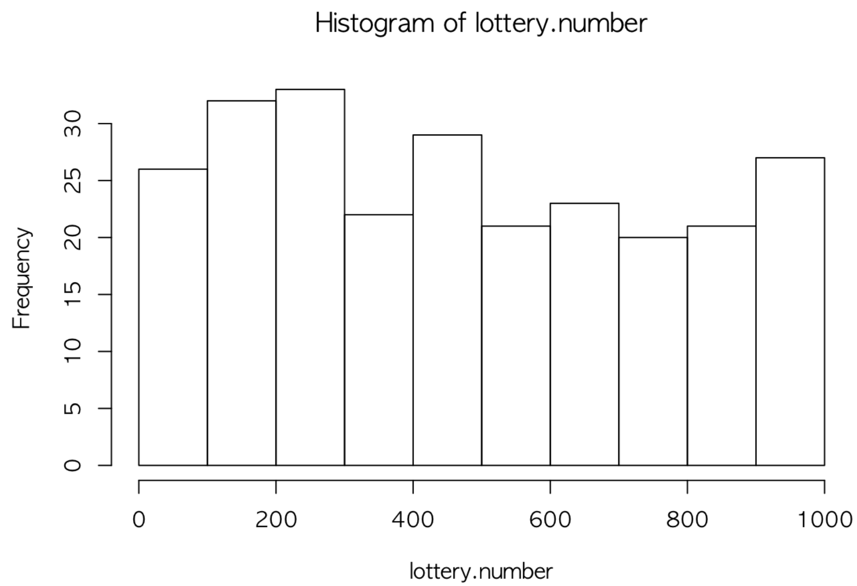
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     230     440     472     734     999
```

```
sd(lottery.number)
```

```
## [1] 294
```

- lottery.number 의 분포를 살피기 위하여 히스토그램 작성

```
h10<-hist(lottery.number)
```



- 각 계급에 관찰된 당첨번호의 갯수를 파악하기 위하여 h10$counts 출력

```
h10$counts
```

```
##  [1] 26 32 33 22 29 21 23 20 21 27
```

- 당첨번호의 갯수가 uniform 하게 추출된 것으로 보아도 무방한지 χ^2 테스트 수행. 왜 다른 argument들을 설정하지 않아도 되는지 help 파일로 확인하고, 이어서 계산되는 값들 중에서 기대돗수 확인.

```
chisq.test(h10$counts)
```

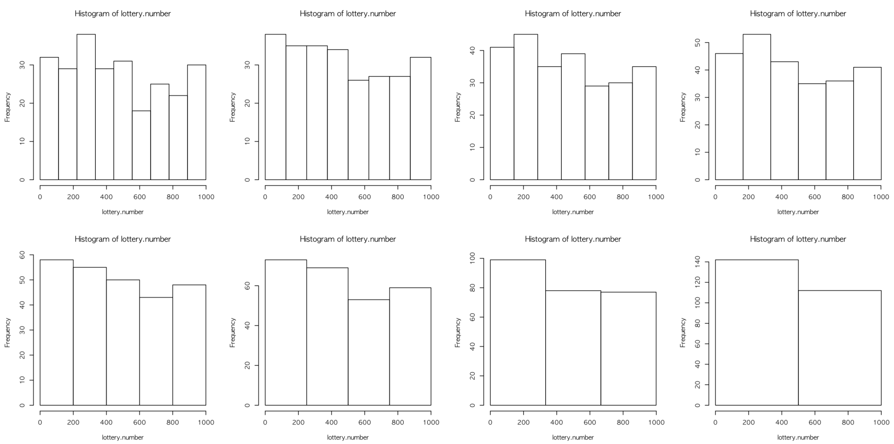
```
##
##  Chi-squared test for given probabilities
##
## data:  h10$counts
## X-squared = 7.97, df = 9, p-value = 0.5373
```

```
chisq.test(h10$counts)$expected
```

```
##  [1] 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4
```

- 계급의 갯수를 바꿔가면서 테스트, breaks로 조정하는 이유에 대하여 생각해 볼 것.

```
opar<-par(no.readonly=TRUE)
par(mfrow=c(2,4))
h9<-hist(lottery.number, breaks=seq(0,999, by=111))
h8<-hist(lottery.number, breaks=seq(0,1000, by=125))
h7<-hist(lottery.number, breaks=seq(0,1001, by=143))
h6<-hist(lottery.number, breaks=seq(0,1002, by=167))
h5<-hist(lottery.number, breaks=seq(0,1000, by=200))
h4<-hist(lottery.number, breaks=seq(0,1000, by=250))
h3<-hist(lottery.number, breaks=seq(0,999, by=333))
h2<-hist(lottery.number, breaks=seq(0,1000, by=500))
```



- 각각의 count 통계량에 대하여 uniformity 적합도 검정을 하기 위하여 다음 수행. 같은 작업을 sapply()로 수행하면 어떤 결과가 나오는지 비교하시오.

```
lapply(list(h9$counts, h8$counts, h7$counts, h6$counts, h5$counts, h4$counts, h3$counts, h2$counts), chisq.test)
```

```
## [[1]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[1L]]
## X-squared = 9.76, df = 8, p-value = 0.282
##
##
## [[2]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[2L]]
## X-squared = 4.52, df = 7, p-value = 0.7183
##
##
## [[3]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[3L]]
## X-squared = 5.55, df = 6, p-value = 0.4753
##
##
## [[4]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[4L]]
## X-squared = 5.28, df = 5, p-value = 0.3832
##
##
## [[5]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[5L]]
## X-squared = 2.73, df = 4, p-value = 0.6036
##
##
## [[6]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[6L]]
## X-squared = 3.95, df = 3, p-value = 0.2666
##
##
## [[7]]
##
```

```
##  Chi-squared test for given probabilities
##
## data:  X[[7L]]
## X-squared = 3.65, df = 2, p-value = 0.1616
##
##
## [[8]]
##
##  Chi-squared test for given probabilities
##
## data:  X[[8L]]
## X-squared = 3.54, df = 1, p-value = 0.05979
```