ASM - Module 1 Exercises

Michal Siwek

Monday, October 12, 2015

Task 1

Data:

```
p_red <- c(2/3, 1/2, 1/3)
p_green <- 1 - p_red
all_lights <- 1:3
```

Calculating the distribution $F_X(x)$. For every $x \in (0, 1, 2, 3)$:

1. find all elementary events that have x red lights
2. calculate the probability of each of those elementary events
3. sum up the probabilities and assign to x

```
FX <- sapply(0:3, function(x) {
    elem_events <- combn(all_lights, x)
    p_elem_events <- apply(elem_events, 2, function(red_lights) {
            green_lights <- setdiff(all_lights, red_lights)
            prod(p_red[red_lights], p_green[green_lights])
        })
    c(x, sum(p_elem_events))
})
rownames(FX) <- c("x", "F(x)")
FX
```

```
##           [,1]      [,2]      [,3]      [,4]
## x    0.0000000 1.0000000 2.0000000 3.0000000
## F(x) 0.1111111 0.3888889 0.3888889 0.1111111
```

Task 2

Data:

```
mn <- 100
sd <- 15
```

a) Probability that $x > 130$

```
pnorm(130, mean = mn, sd = sd, lower.tail = F)
```

```
## [1] 0.02275013
```

b) Probability that $x \in [100, 120]$

```r
pnorm(120, mean = mn, sd = sd) - pnorm(100, mean = mn, sd = sd)
```

```
## [1] 0.4087888
```

Task 3

Data:

```r
pxy <- matrix(
        c(c(.1, .1, 0),
          c(.2, .2, .1),
          c(.1, .1, .1)),
        nrow = 3,
        byrow = T,
        dimnames = list(X=c(0, 1, 2), Y=c(-1, 0, 1))
    )
pxy
```

```
##    Y
## X    -1   0   1
##   0 0.1 0.1 0.0
##   1 0.2 0.2 0.1
##   2 0.1 0.1 0.1
```

a) Marginal distribution p_x for X:

```r
px <- apply(pxy, 1, sum)
x <- as.numeric(names(px))
px
```

```
##   0   1   2
## 0.2 0.5 0.3
```

Marginal distribution p_y for Y:

```r
py <- apply(pxy, 2, sum)
y <- as.numeric(names(py))
py
```

```
##  -1   0   1
## 0.4 0.4 0.2
```

b) Calculating $P(X > 2Y)$:

```r
cond <- matrix(nrow = 3, ncol = 3)
for(i in seq_len(nrow(pxy)))
    for(j in seq_len(ncol(pxy)))
        cond[i,j] <- as.numeric(rownames(pxy)[i]) > 2 * as.numeric(colnames(pxy)[j])
sum(pxy[cond])
```

```
## [1] 0.7
```

c) Calculating covariance of X and Y:

```
mn_x <- sum(x * px)
mn_y <- sum(y * py)
tmp <- matrix(nrow = 3, ncol = 3)
for(i in seq_len(nrow(pxy)))
    for(j in seq_len(ncol(pxy)))
        tmp[i, j] <- pxy[i,j] * (x[i] - mn_x) * (y[j] - mn_y)
cov_xy <- sum(tmp)
cov_xy
```

```
## [1] 0.12
```

Calculating correlation of X and Y:

```
var_x <- sum(px * (x - mn_x)^2)
var_y <- sum(px * (y - mn_y)^2)
cor_xy <- cov_xy/sqrt(var_x * var_y)
cor_xy
```

```
## [1] 0.2250967
```

The variables aren't independent as covariance and correlation are both different than 0. The frequency table for the joint distribution of X and Y should be filled with equal values (i.e. 1/9 here) for X and Y to be independent.

d) Conditional distribution of X given $Y = -1$:

```
FX_y <- sapply(pxy[, 1], function(p) p / sum(pxy[, 1]))
FX_y
```

```
##    0    1    2
## 0.25 0.50 0.25
```

Conditional distribution of Y given $X = 0$:

```
FY_x <- sapply(pxy[1, ], function(p) p / sum(pxy[1, ]))
FY_x
```

```
##  -1   0   1
## 0.5 0.5 0.0
```

Task 4

Data:

```
x_min <- 1.4
x_max <- 1.8
f_x <- 1 / (x_max - x_min)
```

Deriving population mean and standard deviation from the definitions for continuous distributions:

```
mn_x <- 2.5 * x_max^2 / 2 - 2.5 * x_min^2 / 2
E_of_squared_x <- 2.5 * x_max^3 / 3 - 2.5 * x_min^3 / 3
var_x <- E_of_squared_x - (mn_x)^2
sd_x <- sqrt(var_x)
```

We use CLT for sums of random variables to find the mean and standartd deviation for the sum of 50 randoms:

```
n <- 50
mn <- n * mn_x
sd <- n * sd_x
mn
```

```
## [1] 80
```

```
sd
```

```
## [1] 5.773503
```

According to CLT, our random variable (approximate time of 50km cycling) has approximately normal distribution with the mean 80 and standard deviation 5.7735027.

Task 5

Data:

```
n <- 50
mn <- 28.40
sd <- 4.75
```

The confidence interval for population mean using CLT:

```
mn + c(-1, 1) * qnorm(.975) * sd / sqrt(n)
```

```
## [1] 27.08339 29.71661
```

Task 6

Data:

```
n <- 400
rec <- 79
```

Two sided test, at $\alpha = 5\%$:

- $H_0 : \mu = 25\%$
- $H_1 : \mu \neq 25\%$

Assuming the distribution is binomial we can derive standard deviation form the mu.

```
mu0 <- .25
mn <- rec / n
var0 <- mu0 * (1 - mu0) / n
```

Test statistic and p-value:

```
t.stat <- (mn - mu0) / sqrt(var0)
2 * pnorm(t.stat)
```

```
## [1] 0.01531382
```

H_0 is rejected as the p-value is below 5%.

Task 7

Data:

```
mu0 <- 3.2
n <- 50
mn <- 3.05
se <- 0.34
```

Test statistic and the p-value:

```
t.stat <- sqrt(n) * (mn - mu0) / se
pt(t.stat, n - 1)
```

```
## [1] 0.001515951
```

H_0 rejected as p-value is lower than 5%.