

Lab2

Dahun SEO at BIBS Lab.

2021-03-11

Random number generation

배운 내용을 실습하거나 새로운 발견을 확인하는 과정에서 시뮬레이션은 중요한 역할을 합니다. 이번 실습 시간에서는 R의 함수를 이용해서 난수를 생성해보도록 하겠습니다. 사실 R의 함수를 이용해서 발생시킨 난수들은 대개 Uniformly distributed random number를 바탕에 두고 있습니다. 그래서 균등분포를 따르는 난수를 생성하는 알고리즘이 중요합니다. 자세한 내용은 Random number generation 을 읽어보시면 도움이 될 것입니다.

source: Wikipedia

Relating Functions

자주 나오는 확률 변수들과 관련 함수들은 아래와 같습니다.

1. 연속형 확률 변수
 - 정규 분포 : `rnorm`
 - t분포 : `rt`
 - F분포 : `rf`
 - 연속 균등 분포 : `runif`
 - 지수 분포 : `rexp`
 - 감마 분포 : `rgamma`
 - 카이제곱 분포 : `rchisq`
2. 이산형 확률 변수
 - 이항 분포 : `rbinom`
 - 기하 분포 : `rgeom`
 - 초기하 분포 : `rhyper`
 - 음이항 분포 : `rnbinom`
 - 포아송 분포 : `rpois`

Ex) $N(0, 2)$ 난수 5개 추출

```
rnorm(n = 5, mean = 0, sd = sqrt(2))
```

```
## [1] -0.03125426 -1.90519441 -1.01596696 0.08330767 1.41550308
```

분포앞에 붙은 r을 d, p, q 로 바꾸면 기능이 달라집니다. `rnorm`, `dnorm`, `pnorm`, `qnorm` 을 자유롭게 활용할 수 있으면 좋습니다. 예제로 $N(0, 1)$ 의 확률 밀도 함수를 그리고 오른쪽 꼬리 확률 0.05 영역을 근사적으로 시각화해보도록 하겠습니다.

Ex) when $\alpha = 0.05$, $Z_{1-\frac{\alpha}{2}} = 1.96$

```
qnorm(0.975)
```

```
## [1] 1.959964
```

```
qnorm(0.025, lower.tail = FALSE)
```

```
## [1] 1.959964
```

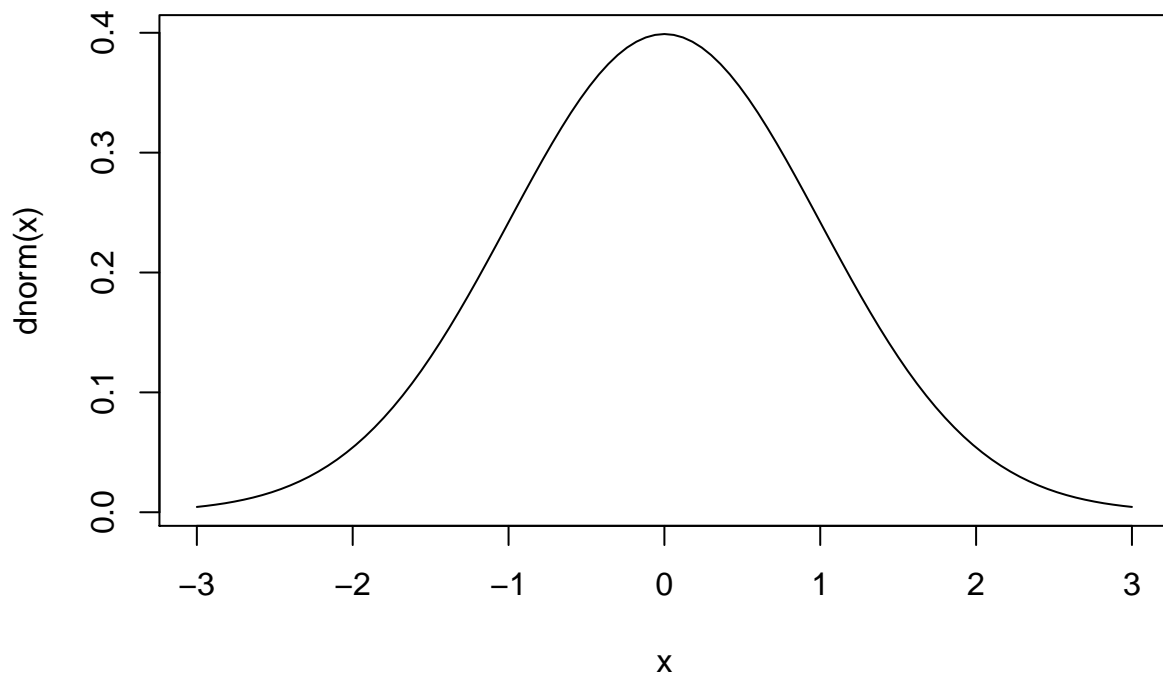
pnorm calculates cdf.

```
pnorm(qnorm(0.975))
```

```
## [1] 0.975
```

curve함수를 사용하면 때때로 간단하게 시각화 할 수 있습니다.

```
curve(dnorm(x), -3, 3)
```



polygon함수는 2차원 평면의 점들을 다각형 형태로 이어 영역을 만드는 함수입니다.

```
x = seq(-3, 3, by = 0.01)
```

```
y = dnorm(x, mean = 0, sd = 1)
```

```
plot(x, y, type = 'l')
```

```
z_0.95 = qnorm(0.95)
```

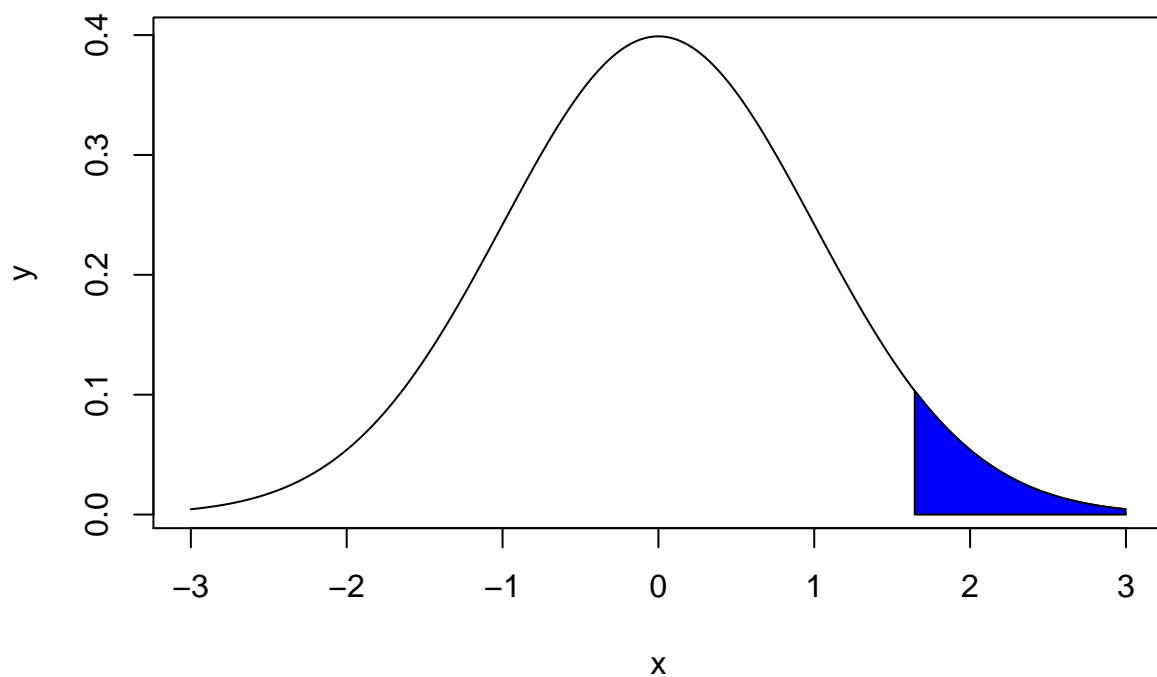
```
z = seq(z_0.95, 3, by = 0.01)
```

```
f.z = dnorm(z, mean = 0, sd = 1)
```

```
z = c(z_0.95, z, 3)
```

```
f.z = c(0, f.z, 0)
```

```
polygon(z, f.z, col = 'blue')
```



Simple Linear regression

수업 시간에 배운 simple linear regression의 LSE를 구해보도록 하겠습니다.
speed : 속도, dist : 제동거리

```
data(cars)
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

$$dist = \beta_0 + \beta_1 speed + \epsilon$$

```
fit = lm(dist ~ speed, data = cars)
summary(fit)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601  0.0123 *
## speed        3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

$$\beta_0^{LSE} = -17.58$$

$$\beta_1^{LSE} = 3.93$$

이제 lm함수를 쓰지 않고 직접 계산해봅시다.

```
beta1 = sum((cars$speed - mean(cars$speed)) * cars$dist) / sum((cars$speed - mean(cars$speed))^2)
beta0 = mean(cars$dist) - beta1 * mean(cars$speed)

cat('beta0 : ', beta0, 'beta1 : ', beta1, '\n')
```

```
## beta0 : -17.57909 beta1 : 3.932409
```

이제 2차원 평면에 그림을 그려보도록 하겠습니다.

```
plot(cars$speed, cars$dist)
abline(a = beta0, b = beta1, col = 'blue')
```

