

PH525x Series - Biomedical Data Science 01

PH525x Series - 바이오의학 데이터 과학 01

실제 주피터 노트북 파일을 배포하진 않겠습니다, R이 꼬여서 마크다운으로만 배포합니다

- How To install R? (R을 어떻게 설치하나요?)

여기서는 윈도우, 리눅스(우분투), 맥의 설치방법에 관하여 이야기할 것입니다.

우선, 밑의 링크를 들어갑시다 (한국 미리 입니다만, 어딜 들어가나 같습니다)

[Cran Nexr](#), [서울대 보건대학원](#), [UNIST Genome Inst.](#)

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

여기에서 각 운영체제에 따라서 윈도우, 리눅스, 맥 을 클릭하여 설치 도구를 다운로드 한 후에, 적절한 지시에 따르면 됩니다.

우분투는 터미널을 여신 후에 이 커맨드 한방이면 설치됩니다

```
sudo apt-get install r-base
```

R // R을 입력하면 R 커맨드라인 창이 나올 것입니다.

- R Studio 설치 (무조건은 아니지만, 정신건강에 이롭습니다)

[여기](#)를 클릭하여 접속하면, R스튜디오를 다운로드받아 설치할 수 있는 창이 나옵니다, 맨 왼쪽에 있는 R-Studio 를 클릭하여 다운로드 하시면 됩니다.

- 주피터 노트북에 커널 설치하기

```
// Anaconda 혹은 Miniconda 를 설치하세요
install.packages(c('repr', 'IRdisplay', 'crayon', 'pbdZMQ', 'devtools'))
devtools::install_github('IRkernel/IRkernel')
IRkernel::installspec() # to register the kernel in the current R installation
```

```
1: In install.packages("swirl") :
  패키지 'curl'의 설치가 0이 아닌 종료상태를 가졌습니다
2: In install.packages("swirl") :
  패키지 'openssl'의 설치가 0이 아닌 종료상태를 가졌습니다
3: In install.packages("swirl") :
  패키지 'httr'의 설치가 0이 아닌 종료상태를 가졌습니다
4: In install.packages("swirl") :
  패키지 'RCurl'의 설치가 0이 아닌 종료상태를 가졌습니다
5: In install.packages("swirl") :
  패키지 'swirl'의 설치가 0이 아닌 종료상태를 가졌습니다
```

- Conda install 명령어를 통한 주피터에 커널 추가하는 방법

```
conda install -c r r-essentials
```

Hello World R!

```
print("Hello World!")
```

```
// 헬로우 워월드!가 정상적으로 출력됨.
[1] "Hello World!"
```

help() function을 이용한 도움말 얻기

```
install.packages('swirl') // swirl 패키지를 설치하라는 명령어.
?install.pakcages //앞에 Questionmark(물음표)를 붙여서 할 수도 있고...
help(install.packages)//help() Function 을 이용하여 도움말을 불러오는 방법도 있다
// 따로 코드 실행한 것은 첨부하지 않겠습니다, 너무 길어요(...)
```

rafalib, downloader 모듈을 설치하고 library() 함수를 이용하여 import 해보기

```
install.packages('rafalib') // rafalib 을 설치합니다
install.packages('downloader') // downloader 을 설치합니다
```

getwd(), setwd() 함수를 이용하여 Working PATH 를 찾고 설정 해보기

```
getwd()
```

[1] "/home/joe" // 홈 디렉토리에 joe라는 폴더에서 시작된다는걸 알 수 있음.

```
setwd("당신이 원하는 시작 디렉토리 PATH")
```

Opt.1 :: Working Dir. 에다가 파일 실습할 파일 다운로드 받아보기.

이 [깃허브 레파지토리](#)에 접속하여, FemaleMiceWeights.csv 라는 파일을 다운로드.

아니면 이 깃 자체를 클론해서 사용해도 됨 `git clone` 깃허브_주소 와 같이 사용하면 깃 자체를 클로닝 할 수 있음, 일단 git 이 설치되어 있어야 함.

깃 설치방법 / 간단사용방법

- 깃은 리누스 토발즈가 svn 에 화가 나서 버전관리 프로그램을 개발하였는데, 그것이 Git 이다.
- 일단 깃을 설치하는 방법은.
 - 윈도우: [Git-scm.com](#)에 접속하여 Git을 다운로드하여 설치도구의 설명에 따라 설치
 - 리눅스: `sudo apt-get install git` 을 입력하면 설치됨
- 클로닝 해보기

```
git clone https://github.com/genomicsclass/dagdata/tree/master/inst/extdata // ex  
tdata라는 이름의 폴더로 클론될 것임.
```

- 깃 로그 보기

```
git log // 내가 보고싶은 깃의 폴더로 들어가서 저 커맨드를 입력하면 로그를 볼수 있음.
```

- git commit 해보기

```
git add .
```

```
git commit -m "커밋 메시지를 입력"
```

- 내 저장소에 동기화 해보기

```
git push origin master // 이후에 이메일 / 비밀번호 입력
```

Clone It!

```
git clone https://github.com/genomicsclass/dagdata // dagdata 라는 폴더를 만들어져  
그 속에 클로닝 될 것임
```

파일이 정상적으로 다운로드 된것을 확인할 수 있음

Opt.2 :: R에서 파일 다운로드 받아보기 :: 추천

```
library(downloader) ## 위에서 install.packages()함수를 사용하여 downloader을 설치  
url <- "https://raw.githubusercontent.com/genomicsclass/dagdata/master/inst/extda  
ta/femaleMiceWeights.csv"  
filename <- "femaleMiceWeights.csv"  
download(url, destfile=filename) // 다운로드 함수를 사용하여 다운로드하는데, destifi  
le 옵션에 filename변수의 값과 같도록 설정.
```

파일을 읽어봅시다!

```
dat <- read.csv(filename) //csv 파일을 읽을 수 있도록 하는 함수 read.csv() 함수를 사  
용하여 파일을 읽어들이기
```

Devtools 설치해보기

과연 설치가 될것인가

```
install.packages("devtools")
```

윈도우는 Rtools 를 다운로드받아서 꼭 관리자 권한 으로 열어서 설치하신 후에 저 위의 명령어
를 실행하시기 바랍니다!

Exercises 01

```
getwd()
```

```
‘/home/joe/dagdata/inst’
```

```
setwd('/home/joe/dagdata/inst/extdata')
```

```
getwd()
```

```
‘/home/joe/dagdata/inst/extdata’
```

CSV 파일을 읽어서 CSV데이터를 출력

femaleMiceWeights.csv 파일을 읽어서 출력해보기

```
dat <- read.csv('femaleMiceWeights.csv')
```

```
dat
```

	Diet	Bodyweight
1	chow	21.51
2	chow	28.14
3	chow	24.04
4	chow	23.45
5	chow	23.68
6	chow	19.79
7	chow	28.4
8	chow	20.98
9	chow	22.51
10	chow	20.1
11	chow	26.91
12	chow	26.25
13	hf	25.71
14	hf	26.37
15	hf	22.8
16	hf	25.34
17	hf	24.97
18	hf	28.14
19	hf	29.58
20	hf	30.92
21	hf	34.02

22	hf	21.9
23	hf	31.53
24	hf	20.73

12번째에 있는 값을 출력해라

처음에는 `dat[12]`만 썼다가 안되서 찾아봤더니 12번째의 모든 행을 출력하려면 12옆에 `[,]` 마크를 붙이면 됨.

```
dat[12,]
```

	Diet	Bodyweight
12	chow	26.25

달라심볼을 이용하여 모든 쥐의 몸무게를 출력하고 11번째에 있는 쥐의 몸무게를 출력

You should have learned how to use the `$` character to extract a column from a table and return it as a vector.

Use `$` to extract the weight column and report the weight of the mouse in the 11th row.

```
dat$Bodyweight
```

1. 21.51
2. 28.14
3. 24.04
4. 23.45

5. 23.68
6. 19.79
7. 28.4
8. 20.98
9. 22.51
10. 20.1
11. 26.91
12. 26.25
13. 25.71
14. 26.37
15. 22.8
16. 25.34
17. 24.97
18. 28.14
19. 29.58
20. 30.92
21. 34.02
22. 21.9
23. 31.53
24. 20.73

```
dat$Bodyweight[11]
```



```
## 답은 26.91
```

length() 함수를 이용하여 여기 있는 모든 쥐의 갯수를 Count 해보기

```
length(dat$Diet)
## 답은 24
```

hf (high fet) 에 있는 쥐들을 출력하고, 그들의 몸무게의 평균을 구해보기

```
library(dplyr) ## library() 함수로 dplyr 불러와서...
dat <- read.csv('femaleMiceWeights.csv')
hf <- filter(dat, Diet=="hf")
hf
```

아웃풋 :

	Diet	Bodyweight
1	hf	25.71
2	hf	26.37
3	hf	22.80
4	hf	25.34
5	hf	24.97
6	hf	28.14
7	hf	29.58
8	hf	30.92
9	hf	34.02
10	hf	21.90
11	hf	31.53
12	hf	20.73

```
hf$Bodyweight
## [1] 25.71 26.37 22.80 25.34 24.97 28.14 29.58 30.92 34.02 21.90 31.53 20.73
mean(hf$Bodyweight)
## [1] 26.83417
```

dplyr 모듈이란 무엇인가?

Hadley Wickman가 작성한 데이터 처리에 특화된 R Package이다, C++ 로 만들어져 빠른 속도를 나타낸다는게 특징

R에는 데이터를 전처리할수 있는 함수들이 지원되지만, 이런 외부 함수의 도움을 받으면 더 빠르고 더 쉽게 할 수 있다는 장점이 있다.

dplyr 모듈을 이용하여 chow만 출력해 보기

```
library(dplyr) # 라이브러리 함수를 이용하여 dplyr 라이브러리를 부르기.
dat <- read.csv("femaleMiceWeights.csv") # read.csv() 함수를 이용해서 csv파일 읽기
chow <- filter(dat, Diet=="chow") # 필터 함수를 이용하여 Diet 항목에서 chow만 출력해보기
```

select() 함수를 이용하여 특정 값의 값을 출력해보기

```
chowVals <- select(chow, Bodyweight)
#아까 생성했던 chow 변수를 넣고, 그 뒤에 출력할 Bodyweight 를 넣음
```

빠이쁘(pipe)

아마 리눅스와 비슷한 동작을 하는 것 같습니다 (|)

%>% 심볼을 사용합니다

```
chowVals <- filter(dat, Diet=="chow") %>% select(Bodyweight)
```

와 같이, 필터했던 정보를 select() 함수에 넘겨줌과 같은 것을 처리할 수 있습니다.

ExerCises 02

msleep_ggplot2.csv 파일을 읽어서 type을 출력

```
dat <- read.csv('msleep_ggplot2.csv') # read.csv() 함수로 파일을 읽어서...
class(dat) # [1] "data.frame" 을 리턴
```

Primates(영장류)인 동물들은 몇마리인지 Count 하기

```
Value <- filter(dat, order=="Primates")
length(Value) # 12마리
```

Pipe를 이용하여 영장류들의 총 수면시간 출력

```
Value <- filter(dat, order=="Primates") %>% select(sleep_total)
```

RESULT

	sleep_total
1	17.0
2	10.0
3	10.9
4	9.8
5	8.0
6	9.5
7	10.1
8	11.0
9	9.7
10	9.4
11	11.0
12	9.6

mean() 함수로 영장류의 평균값 출력

```
mean(Value$sleep_total)
# 출력: [1] 10.5
```

Math Notations

Introduction to Random Variables

~~페이퍼를 참고하라고 했는데, 없어서 크게 당황하여 넘어가도록 하겠습니다~~

± 값과 $P < 0.001$ 에서 ± 의 값과 P값은 뭘 뜻하는 걸까요?

여기에서는 쥐 데이터베이스를 이용하여 P값과 ± Value 에 대하여 알아보고, 그것을 R로 구현해

보도록 하겠습니다.

```
dat <- read.csv("femaleMiceWeights.csv") ## read.csv() 함수를 이용하여 CSV파일을 읽어들입니다
```

```
head(dat) ## head() 함수를 이용하여 상위 6개 데이터를 뽑아봅니다
```

	Diet	Bodyweight
1	chow	21.51
2	chow	28.14
3	chow	24.04
4	chow	23.45
5	chow	23.68
6	chow	19.79

각 그룹에서의 평균값을 구해보도록 하겠습니다, 전에 사용했던 dplyr 모듈을 사용할 것입니다.

```
library(dplyr)
control <- filter(dat,Diet=="chow") %>% select(Bodyweight) %>% unlist
treatment <- filter(dat,Diet=="hf") %>% select(Bodyweight) %>% unlist
## unlist 함수는 기존에 있던 것을 numeric 으로 바꾸는 함수.
print(mean(treatment)) ## treatment의 평균값을 나타냄
## [1] 26.83417
```

```
print(mean(control))
## [1] 23.81333
```

무거운 쥐와, 가벼운 쥐의 평균값을 찾아서 그들을 뺌으로서 얼마나 더 무거운지 구해보도록 합니다.

```
obsdiff <- mean(treatment) - mean(control)
print(obsdiff)
## [1] 3.020833
```

Random Variables (확률변수)

```
population <- read.csv("femaleControlsPopulation.csv") ## 파일을 읽어서.
population <- unlist(population) ## Numeric 값으로 바꿈.
```

그런 후에 무작위로 12마리의 쥐를 잡아서 population 에다가 집어넣고 평균값을 알아내 보도록

하겠습니다.

~~천하, 우리에게 12마리의 쥐가 있습니다~~

```
control <- sample(population,12) ## 12마리의 쥐를 집어넣고.  
mean(control) ## 평균을 뽑아보면...  
  
## 결과값은 [1] 24.11333, [1] 24.40667, [1] 23.84 로 나온다!
```

Null Hypotesis (귀무가설)

뜻: 두 집단을 대상으로 실험을 했을 때 각 집단에서 동일한 결과가 나올 것이라는 가설.

```
control <- sample(population,12) ## 일단 12마리 랜덤으로 뽑고  
treatment <- sample(population,12) ## 그 다음에 12마리를 랜덤으로 또 뽑은 후에  
print(mean(treatment) - mean(control)) ## 평균에서 평균을 뺀 값은...  
## [1] 0.5575
```

그러면 이것을 한번 1000번을 해봅시다, 여기에서 같은 Percentage 가 얼마나 나올지 알아보는 것입니다, 여기에서는 for-loop 을 사용하여 루프를 돌립니다.

```
n <- 10000  
null <- vector("numeric",n)  
for (i in 1:n) {  
  control <- sample(population,12)  
  treatment <- sample(population,12)  
  null[i] <- mean(treatment) - mean(control)  
}
```

1000번의 루프 후에 우리가 계산한 값이 obsdiff 보다 몇 퍼센트 큰지 한번 알아보시다.

```
mean(null >= obsdiff) ## null 이 obsdiff보다 크거나 같은 것들  
## [1] 0.0138
```

1.5%만이 다이어트 효과가 없다는 이야기이고, 이 책에서 P-Value값에 대하여 알아볼 것이다.

Distributions (분포)

저 위에서 **Null Hypotesis** 에 대하여 배워왔고, 이제부터 **분포**에 대하여 알아보도록 하겠습니다.

여기에서는 **UsingR** 이라는 데이터셋을 가지고 실습해보도록 하겠습니다

```
library(UsingR)
x <- father.son$fheight ## 1078개의 키 데이터셋을 불러오기
round(sample(x,10),1)
## [1] 67.4 64.9 62.9 69.2 72.3 69.3 65.9 65.2 69.8 69.1
```



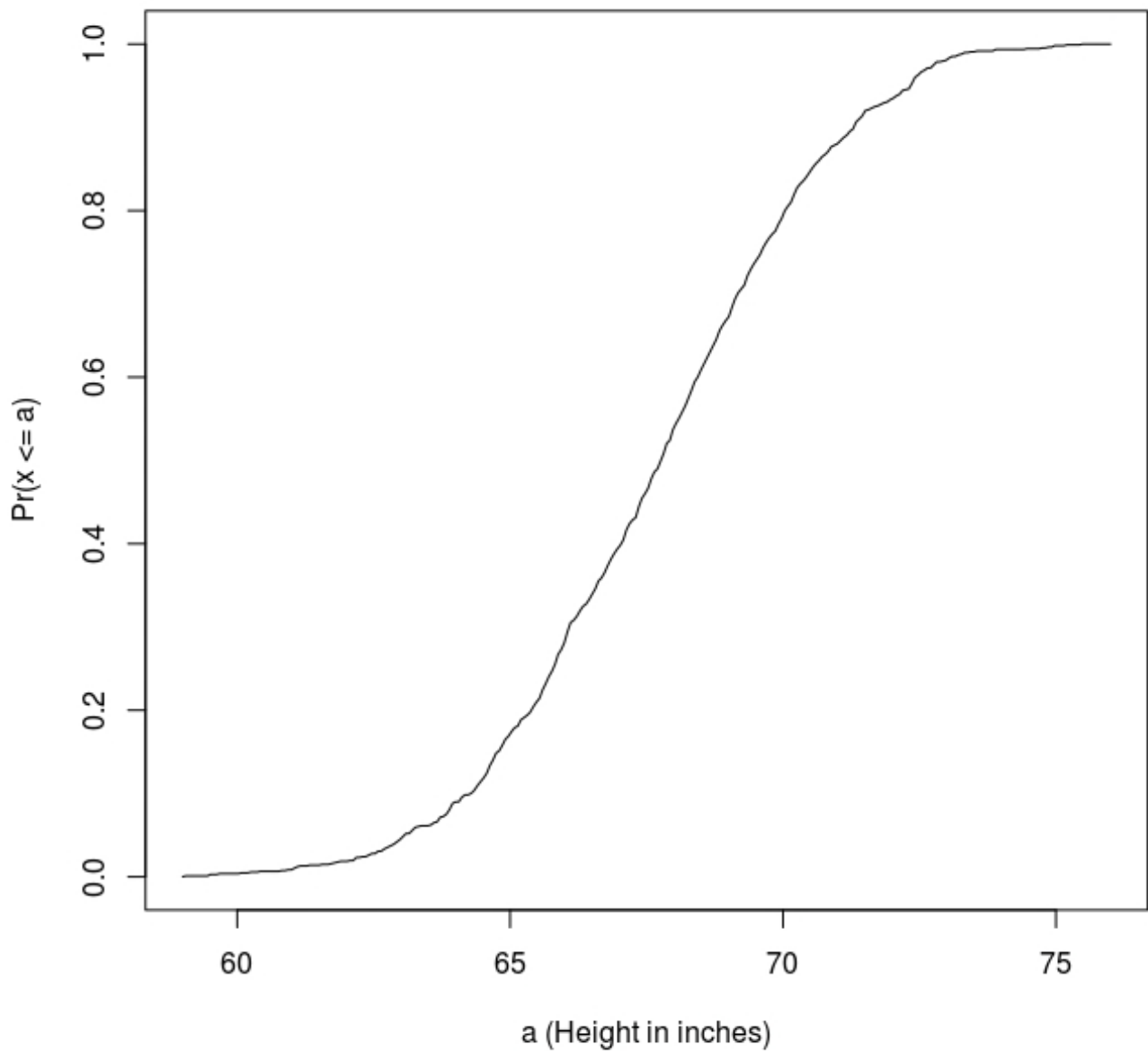
1이 무슨일을 하는건가 싶어서 2를 집어넣고 3을 집어넣어 봤더니, 소숫점 자릿수 역할을 하는 거 같음.

Comulative Distribution Function. (누적표본함수)

누적표본함수: 실수값을 가지는 확률변수 x 의 확률적 분포를 표현하기 위해 이용한다.

$$F(x) = P(X \leq Qx)$$

```
smallest <- floor( min(x) ) ## 작은 수를 밑에다가 놓고
largest <- ceiling( max(x) ) ## 큰 수를 위에다가 놓고
values <- seq(smallest, largest,len=300) ## 작은수, 큰수, 300개까지 지정해준 후에
heightcdf <- ecdf(x) ## x의 ecdf 값을 hightecdf 값을 넣고 (경험분포함수)
plot(values, heightcdf(values), type="l", xlab="a (Height in inches)",ylab="Pr(x
  <= a)")
## plot를 그리면..
```

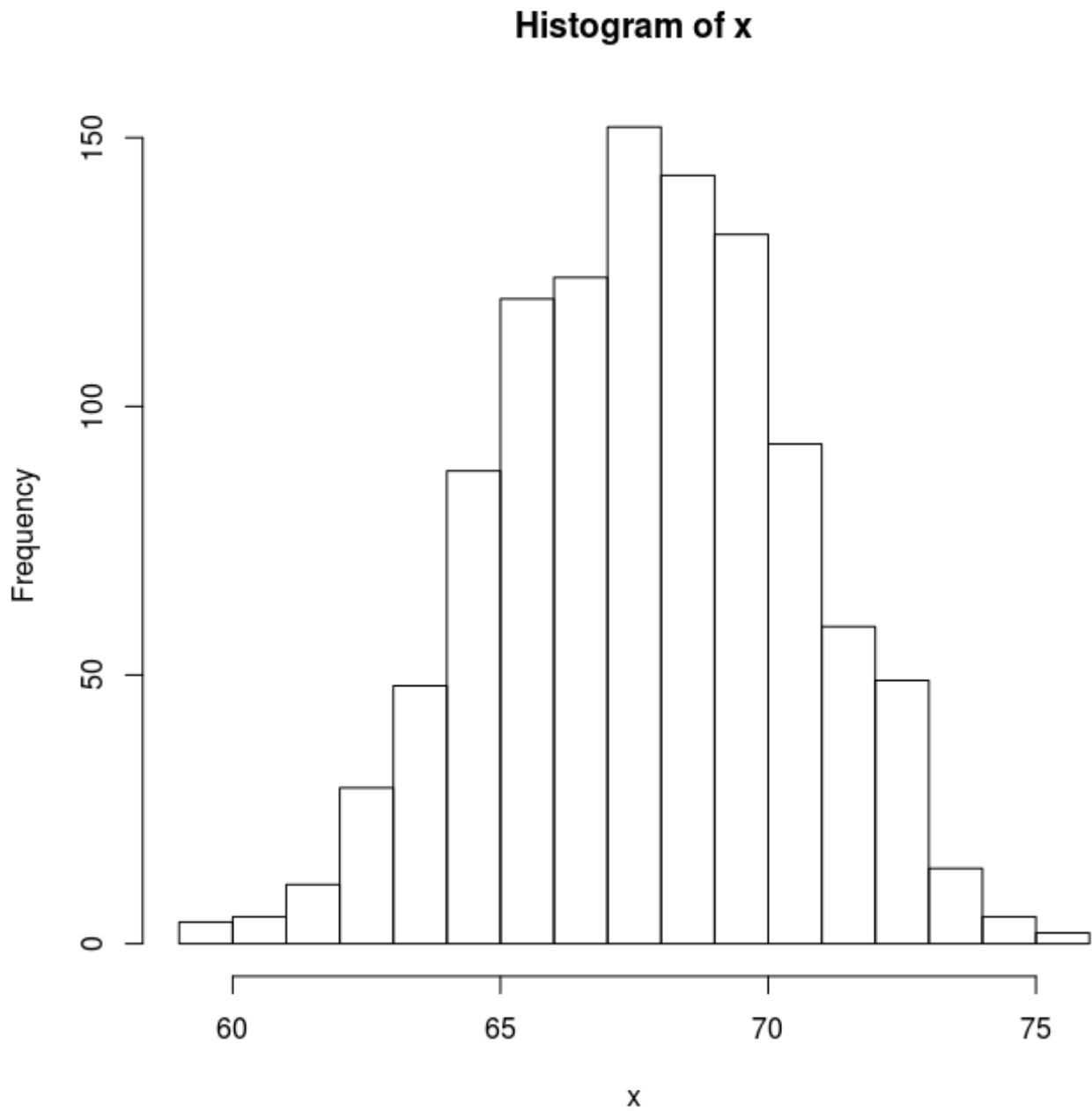


히스토그램

위에서 했던 값을 가져와서 히스토그램으로 만들어 보겠습니다, 여기에서는 `hist()` 함수를 사용하면 됩니다

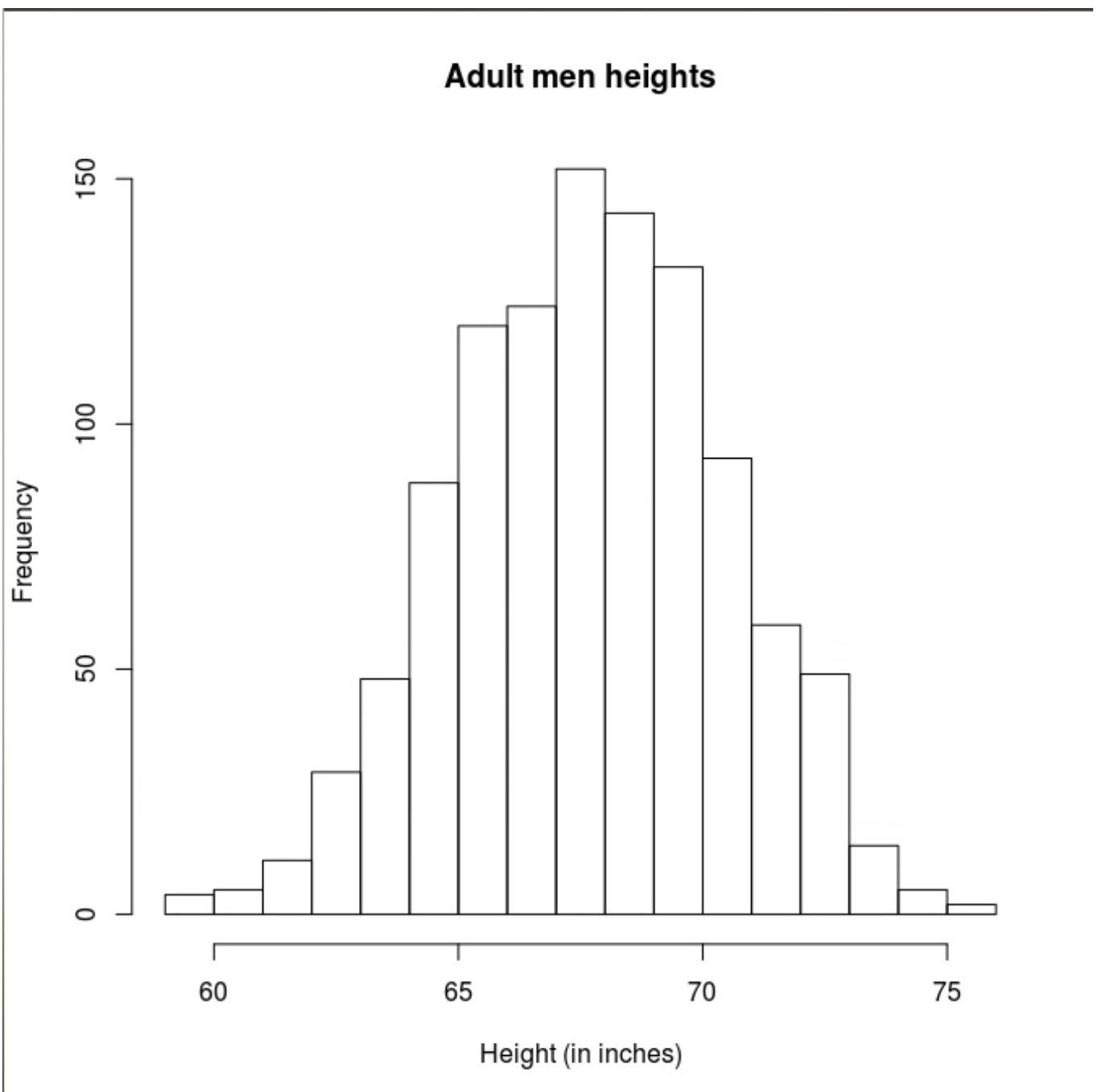
```
hist(x)
```

```
## 히스토그램으로 만들어주는 함수에다가 x값을 집어 넣으면...
```



근데 옆에 설명을 추가하고, 좀더 멋지게 만들려면 아래의 코드를 추가하면 됩니다.

```
bins <- seq(smallest, largest)
hist(x,breaks=bins,xlab="Height (in inches)",main="Adult men heights")
```

Probability Distribution (확률 분포)

$$\Pr[a \leq X \leq b] = \int_a^b f(x) dx$$

```
n <- 100
library(rafalib)
nullplot(-5,5,1,30, xlab="Observed differences (grams)", ylab="Frequency")
totals <- vector("numeric",11)
for (i in 1:n) {
  control <- sample(population,12)
  treatment <- sample(population,12)
```

```

nulldiff <- mean(treatment) - mean(control)
j <- pmax(pmin(round(nulldiff)+6,11),1)
totals[j] <- totals[j]+1
text(j-6,totals[j],pch=15,round(nulldiff,1))
##if(i < 15) Sys.sleep(1) ##You can add this line to see values appear slowly
}

```

코드는 12개씩 랜덤으로 추출하기 때문에 그래프로 뽑아나올 때의 결과값이 다릅니다.

Populations, Samples and Estimates (밀도, 샘플 그리고 예상)

밀도 파라미터

첫번째로 하여야 할 것은, 당신이 어느 분야의 밀도에 관심이 있는지를 알아보아야 한다.

쥐들의 몸무게를 예로 들자면, 우리는 두개의 밀도를 예로 들 수 있다.

첫번째 예는, Female Mice의 다이어트 - Female Mice 과체중의 다이어트

여기에서 샘플링을 하기 위하여 우리는 “**mice_pheno.csv**” 라는 데이터셋을 사용할 것이다.

여성 쥐 중에서 Chow 단의 쥐들의 Pop.

```

library(dplyr)
dat <- read.csv("mice_photo.csv")
controlPopulation <- filter(dat, Sex == "F" & Diet == "chow") %>% select(Bodyweight) %>% unlist
length(controlPopulation)

```

저들 중에서 과체중 쥐들의 Pop.

```

hfPopulation <- filter(dat, Sex == "F" & Diet == "hf") %>% select(Bodyweight) %>% unlist
length(hfPopulation)

```

평균은...

$$\mu_X = \frac{1}{m} \sum_{i=1}^m x_i \text{ and } \mu_Y = \frac{1}{n} \sum_{i=1}^n y_i$$

변화량은...

$$\sigma_X^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_X)^2 \text{ and } \sigma_Y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \mu_Y)^2$$

중심극한정리 (CLT)

이 [링크](#) 가 설명이 더 잘되어 있는것 같습니다.