통계학 입문

2021-03-21

데이터과학융합스쿨

손 대 순 교수

여러부~ㄴ. 잘 지내고 있나요? 지난 시간에 여러분들에게 R과 R studio를 설치해 보라고 하였습니다. 아주 깜찍한 인증샷도 있었고, 참 기발하다 하는 생각을 가지게 하는 사진도 있었어요. 특히, 강의 교안에서 주어진 예제 뿐만이 아니라, 새로운 데이터를 가지고 해 보는 시도가 너무 좋았습니다. 여러분들이 조금 재미를 가져 주는 것 같아서 많이 뿌듯합니다. ^^

해 보니 어떻던가요? 물론 별 문제 없이, 주어진 과제가 잘 수행되는 사람도 있었겠지만, 뭔가 오류가 생기고, 예상치 못한 화면이 나오고 등등 짜증 나는 상황이 연출된 사람도 많을껄요. 여러분들의 생각을 한 번 짐작해 봅니다.

- 에이~C, 설치하는 화면이 교수님이 준 화면하고 다른데? 뭐야~
- 왜 아무 화면도 안 나오지? 내 컴퓨터가 이상한가?
- 시뻘건 오류가 뜨는데? 뭔 말인지 알 수가 없네. 역시 난 컴퓨터랑 안 친해.

뭐 이런 등등의 반응이겠지요. ㅎㅎ 찔리는 사람 있죠?

통계학 과목이기는 하지만, 딱딱한 이야기는 좀 미루어 두고, 신입생들이 많으니 대학생활을 시작하는 여러분들에게 몇가지 팁을 드릴께요.

첫째, 공부에 대한 개념을 바꾸어야 합니다.

고등학교 때의 수학, 국어, 사회 등과 같이 정해진 과목만이 공부는 아닙니다. 공부를 과목수준에서 정의하면, 나머지 시간은 아무 생각없이 노는 시간이 되겠죠. 대학에서는 실컷 놀 수 있는 기회가 있습니다. 그렇다면 무엇이 공부일까요? 노는 것처럼 보였던 모든 것들이 공부라고 할 수 있습니다. 친구를 만나는 것도, 노래를 부르는 것도, 인터넷 기사를 찾고, SNS를 하는 것도 모두 공부입니다. 이제는 노는 것과 공부의 구분이 모호해 질 것입니다. 내가 공부라고 생각하는 것이, 바로 공부입니다. 어떻게 얘기하고 행동하는 것이 더 좋은 친구가 될 것인가 잘 생각해 보는 것은 아주 중요한 공부 중에 하나이겠지요. 다양한 경험들을 해야 합니다. 무엇이 내 마음을 뿌듯하게 하는가를 스스로 찾아야 합니다. 결국 내가 뿌듯해 하는 일을 잘 하도록 목표로 삼아야 합니

다. 그러려면 다양한 경험을 적극적으로 해야겠지요. 해 보지 않고서는 내게 맞는지, 내 스타일이 아닌지 알 수 없습니다. 많은 경우에 해보지도 않고, 내 적성이 아니라고 말합니다. 적성이 아닌 게 아니라, 내 적성인지 알지 못한다가 더 정확한 표현이겠지요. 나를 정확히 알려는 노력을 해야 합니다.

둘째, 컴퓨터와 친해지세요.

게임을 해도 좋습니다. 인터넷을 해도 좋습니다. 어떻게 시작해도 괜찮습니다. 컴퓨터에 게임을 설치해 보세요. 다양한 게임의 종류가 있지만, 어떤 게임은 설치가 매우 어렵습니다. 컴퓨터의 사양에 대한 이해가 필요한 경우도 많죠. 인터넷이 안되는 컴퓨터에서 인터넷 연결도 시도해 보세요. 파워포인트, 엑셀, 한글 등이 가지고 있는 다양한 기능들을 활용해 보는 것도 필요하겠네요. 친구의 컴퓨터에 원격 접속을 한 번 시도해 보세요. 친구가 무엇을 하고 있는지 우리 집에서 볼 수 있습니다. 원격데스크탑 이라는 것도 있죠. 컴퓨터는 친해지면 아주 새로운 경험을 제공합니다. 내가 앞으로 사용할 비서이기도 하고, 좋은 조력자이기도 합니다. 이 녀석을 잘 알아야 합니다. 아는 만큼 할 수 있는 일들이 늘어납니다. 학교에서 잘 가르쳐 주지 않겠느냐구요? 가르쳐 줄 수 있는 지식은 손톱만큼도 안됩니다. 그보다 훨씬 더 많은 것들이 있어요. 강의에서는 여러분들이 전혀 생각지 못했던 분야에 대한 키워드를 알려주는 것이예요. 키워드를 알게 되면 여러분들은 검색도 해 보고, 스스로 실습도 해 보면서 키워드를 중심으로 지식을 쌓아야 합니다. 바로 이것이 고등학교 공부와 크게 다른 점이라고 할 수 있죠. 게임도 인터넷 서핑도 공부인 이유입니다. 이제 컴퓨터는 여러분과 뗄레야 뗄 수 없는 친구입니다. 친한 친구도 있겠지만, 아직 생소한 친구들도 많을 것 같아요. 적극적으로 친해 지세요. 컴퓨터라는 녀석, 부끄러움을 많이 타는지 지가 친해지려는 노력은 별로 안하는 것 같네요. 여러분들이 다가가 주세요.

셋째, 시간관리가 중요합니다.

고등학교 때까지는 노는 것과 공부하는 것이 구분되어 있으니, 특별히 시간 관리라고 할 것이 없습니다. 본인의 본능을 이기고 공부하도록 직/간접적인 강요가 많았겠지요. 이제는 대학 생활입니다. 강의도 들어야 하고, 과제도 있고, 친구들도 만나야 하고, 영화도 봐야 합니다. 동아리 활동도 할 수 있고, 자격증을 따고 싶은 친구도 있습니다. 모두 중요한 일들입니다. 한꺼번에 중요한 일들이 갑자기 많이 생겼습니다. 어떻게 해야 할까요? 시간 관리가 중요합니다. 중요한 일과 중요하지 않은 일, 급한 일과 급하지 않은 일로 나누어 보죠. 그리고는 주 단위로 계획을 세워 보세요. 내가 자유로이 쓸 수 있는 시간이 어느 정도인지도 계산해 보세요. 그 시간들을 내가 잘 쪼개어 사용해야 합니다. 특히, 중요하다고 생각하는 일은 조금씩이라도 매일 진행하는 것이 필요해요. 몰아서 할 수 있는 일과, 매일 조금씩이라도 해야 하는 일들도 구분해 보죠. 일주일 단위라도 시간을 계획적으로 사용하려는 노력은 나중에 큰 차이를 만들 것입니다. 캘린더 앱을 사용하는 것도 권장합니다. 기록해 두지 않으면 잊기 쉽겠죠.

통계학 입문 수업은 여러분이 통계학적인 개념을 가지도록 구성하고 이야기하려고 합니다. 많

은 경우 학습의 효과는 교수자의 역할과 수강자의 역할이 잘 맞물려야 극대화 됩니다. 교수자는 "무엇을" 가르칠 것인가, "어떻게" 가르칠 것인가에 집중해야 합니다. 주어진 환경에서 최대한의 교육효과를 얻기 위해서 고민하고, 연구해야 합니다. 이 부분을 집중하여 이번 학기 수업을 진행하고자 합니다. 학생은 "태도"가 가장 중요 합니다. 학문에 대한 태도를 말합니다. 쉽게 들을 수 있는 내용들이지만, 실상 많은 사람들이 많은 희생을 해 가며 쌓은 지식들입니다. 이를 진심으로 받아들이고 탐구하려는 자세가 매우 중요합니다. 설령 지금은 답을 알고 있을지라도, 당시의 시대 상황과 배경을 고려해 그 입장에서 생각해보고, 고민하는 자세가 반드시 필요합니다. 학습이란 "학(學)"과 "습(習)"이 합쳐진 말임을 잘 생각해 보아야 합니다. 단순히 교수자에게 전달받은 지식은 실상 지식이 아닙니다. "습"의 과정을 거쳐야 비로소 내 것이 되고, 내 스스로 활용 가능한 지식이 되는 것입니다.

이번 과제를 통해 여러분이 위의 내용을 경험한 것입니다. 오류가 생기면 이것이 무엇 때문인지를 찾아가는 과정이 중요한 공부 과정이예요. 여러분들이 짜증을 내는 만큼 컴퓨터도 짜증을 내고 있는 것이거든요. 여러분들이 대인배의 입장에서 컴퓨터가 알아들을 수 있게 잘 달래주는 것이 또한 중요합니다.

아마도, 컴퓨터 이름(여러분들의 계정)이 한글로 되어 있어서 발생하는 문제들이 있었을 것이구요. '관리자 권한으로 실행'해야 했었던 부분도 있었을거예요. 운이 좋게 아무런 문제가 생기지 않았던 친구들도 있겠구요.

학교에서 여러분들에게 가르쳐 줄 수 있는 컴퓨터 사용법이라는 것은 명백히 한계가 있습니다. 도움이 될만한 무언가를 해 보라고 하는 것까지가 교수의 역할인 것 같아요. 실제 해 보면서 발생하는 다양한 문제들을 여러분들이 스스로 풀어가는 과정이 중요한 공부 과정입니다. 또한, 모든 문제들을 교수가 다 알지도 못하구요. 컴퓨터라는 것이 모두 똑같아 보이지만, 사람들 간의 차이만큼이나 서로 다릅니다. 구성하고 있는 부품도 모두 다르고, 운영체계도 다르고, 설치되어 있는 프로그램이나 버전도 거의 다르지요. 사람의 몸을 구성하고 있는 부품이 같은 듯 다르듯이, 컴퓨터도 마찬가지예요. 때문에 무슨 일을 할 때, 모든 컴퓨터가 동일한 반응을 보이는 경우는 흔치 않습니다. 여러분의 컴퓨터는 여러분 각자가 제일 잘 아는 사람이 되어야지요. 필요한 프로그램을 설치할 줄 알고, 문제가 되는 부분을 잘 치료해 줄 수 있어야 합니다. 오류가 생기거나, 컴퓨터에 큰 이상이 발생하는 것을 두려워하지 마세요. 참 희한하게도 컴퓨터는 싹 초기화 하면 새로 산 컴퓨터처럼 떨어졌던 성능이 돌아 온답니다. 문제가 생겼던 부분도 사라지죠. 몇 번이고 새로 시도해 보면 됩니다.

이러한 시도를 다양하게 경험해 본 사람과 그렇지 않은 사람의 차이가 바로 실력 차이예요. 그러니 앞으로는 오류가 생기고, 예기치 않은 문제가 생겼을 때, 환호성을 쳐야 합니다. 공부할 꺼리가 생긴 것이지요. 무엇을 공부해야 할지 모르는 경우가 대부분인데, 이걸 공부하면 된다고 알려 주는 것이니 환호성을 칠만 하죠. ㅋㅋ 약간 또라이 같기는 하겠지만 말이죠.

자, 오늘 수업을 해 봅시다.

'탐색적 자료분석 (Exploratory Data Analysis)' 실습을 해 봅시다

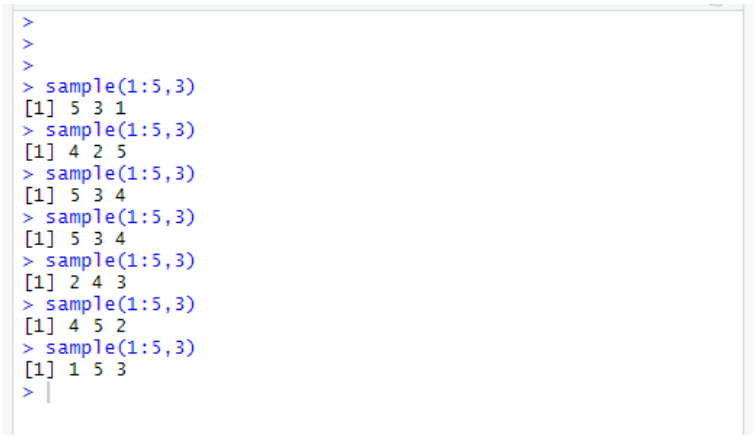
대면수업을 한다면, 이러한 일을 과제물로 주겠지만, 비대면 수업이라 몇가지 과정은 여러분들에게 안내할 수 있어서 좋네요. 나는 할 일이 더 늘었지만...

자, R을 켜세요. (여기서 R은 R studio를 실행시키거나, 그냥 R을 실행시키거나 상관없습니다. R studio는 R을 좀 더 편하게 사용하기 위한 툴이예요. 나중에 고급 프로그래밍을 해야 할 때는 R studio로는 안되는 경우가 있습니다. 특히, 서버에서의 큰 작업을 할 때 그래요. 지금은 초급 단계이니 R studio를 사용해도 괜찮아요. 그냥 R만 실행시켜서 해 보아도 좋습니다.)

> sample(1:5, 3)

이런 명령을 줘 봅시다. 결과가 어떤가요?

똑같은 명령을 다시 실행시켜 보세요. 키보드에서 위쪽 화살표 버튼을 누르면 이전에 실행시킨 명령어가 다시 나옵니다. 여러 번 실행시켜 보세요.



실행 시킬 때 마다 결과가 다르죠?

sample이라는 명령어는 'random sample'을 하라는 명령어입니다.

1에서 5까지의 숫자 중에 3개를 뽑으라는 말이죠. 그러니 매번 실행할 때마다 결과가 다르게 나올 수 밖에요.

다음은 이렇게 명령을 실행시켜 봅시다.

> sample(1:100, 200)

에러가 나지요? 에러가 안 나면 이상한 겁니다. 위의 명령어를 해석해 보면, 1에서 100까지의 숫자 중에서 200개를 뽑으라는 명령어입니다. sample 명령어는 기본적으로 '비복원추출'이예요. 한 번 뽑힌 숫자가 다시 뽑히지 않습니다. 그러니 100개의 숫자 중에 200개를 뽑을 수는 없겠지요. 그러면 이를 '복원추출'하면 가능하겠네요.

> sample(1:100, 200, replace=T)

```
>
> sample(1:100, 200, replace=T)
  [1]  37  32  75  26  73  41  51  37   5   4  99  84  18
 [14]  86  72  61  12  21  51  29  59  43  85   4   4  40
 [27]  91  99  42  42  80  75   9  72  44  60  51   1  28
 [40]   8  24  46  66  88  15   7  30  12  77  92  13  79
 [53]  59  53  63  16  98  87   8  84  76  76  23   3  22
 [66]  20  18  70  81  81  52  40  26  66  31  73  43  82
 [79]  52  40  77  35  89  93  44  57  41  23  80  42  84
 [92]  79  17  24  36  24   2   4  59  30  20  24  71   6
[105]  24  10  51   9  66  94  14  85  46  80  22  87  34
[118]  40  89  31  14  91  69  40  33  44  79  10  88  28
[131]  83  95 100  28  57  18  69  42  56  91  77  16  72
[144]  36  52  72  90   4   2  89  25  10  52  98  10  70
[157]  17  20  99  76  61  27  48  26  62  87  99  33  78
[170]  44   2  26   2  62  55  22  63  88  91  75  23  12
[183]  92   7  14  38  91  36  33  41  48  88  41  67  66
[196]  59  18  27  31  31
>
```

자, 이렇게 나오나요? 나오는 값이 나와 같을 수는 없겠지요. random하게 뽑은 수이니 위의 화면과 같다면 그게 이상한거죠. 아무튼 1~100까지의 숫자 중에서 200개를 뽑았습니다. 얼핏 보니 같은 숫자들이 있지요? 복원추출이니 당연합니다.

자, 이렇게 뽑은 숫자들을 여러분들이 가지고 있는 데이터라고 합시다.

위와 같이 명령어를 주면, '숫자를 뽑는 작업만' 합니다. 이 숫자들을 다시 들여다 보고 싶어도, 이를 가지고 그래프를 그리고 싶어도 할 수가 없습니다. 왜냐 하면, 한 번 뽑아놓기만 하고 어디에 저장해 놓지는 않았거든요.ㅎㅎ 컴퓨터라는 것이 이렇게 단순합니다. 그래서 어디에 저장을 해 놓아야겠어요.

```
>
> example <- sample(1:100, 200, replace=T)
>
>
> example
  [1]  39  69  35  27  15  38  54   4  61   5  61  52  20
 [14]  31  91  63  35  15  52  52  29  13  48  97  58  64
 [27]  10  48  69  60  27 100  94  82  84  49  75  69  57
 [40]  56  16  65  58  47  89  87  36  26  98  49  92  27
 [53]  10  31   6  97  27  94  72  63  77  17  86  56  31
 [66]  22  15  55  31   3  74  47  67  88   6  68  33   4
 [79]  91  29  24  95  85  27  77  21  29  85  25  37  69
 [92]   4  10   3  62  29  46  95  37  28  32  13   2  12
[105]  47  66  92  61  19   2  97  15  20  81  63  23  66
[118]  13  14  55  76  17  52  95  41  42  30  42  10  93
[131]  67  29  88  58  95  77  40  66   7  44  57  58  39
[144]  72  78  28  60  40   5  44   8  31  98  25  88  19
[157]  41  25  50  77  27  97  68  67   4  35  56  73  82
[170]  35  31  64  34  81  67  37  90  20  99  85  88  76
[183]  21  78  46  39  68  98  37  60  42  97  18  79  45
[196]  73  48  87  34  65
>
```

위의 명령어는 샘플을 뽑아서, example이라는 변수에 저장하라는 명령입니다. 이 명령만을 주면 화면에 무엇을 뽑았는지는 보여주지 않지요. 저장하라는 명령어를 주었으니 저장만 할 뿐입니다. example에 무엇이 들어있는지 살펴보기 위해서는 변수 이름을 그냥 쳐 주면 보여줍니다. 다시 뽑았으니, 아까와는 숫자가 달라졌겠지만, example을 여러 번 쳐 보면, 변수에 들어 있는 숫자가 바뀌지는 않는 다는 것을 알 수 있을거예요.

자, 이제 여러분들이 가진 데이터가 있는 상황입니다. 데이터는 example이라는 변수에 들어가 있지요. 대략 이 자료들이 어떻게 생겼는지 한 번 살펴 봅시다.

앞선 강의에서 '도수분포표'라는 것이 있었지요? 몇 개의 계급구간으로 나누어 그 구간 내에 몇 개의 데이터가 있는지를 세어 놓은 표를 말합니다.

우선 계급구간을 나누고 데이터를 각 구간에 넣는 것이 필요하겠지요?

```
>
> interval <- cut(example, breaks=c(0,20,40,60,80,100))
>
> interval
  [1] (20,40]  (60,80]  (20,40]  (20,40]  (0,20]
  [6] (20,40]  (40,60]  (0,20]   (60,80]  (0,20]
 [11] (60,80]  (40,60]  (0,20]   (20,40]  (80,100]
 [16] (60,80]  (20,40]  (0,20]   (40,60]  (40,60]
 [21] (20,40]  (0,20]   (40,60]  (80,100] (40,60]
 [26] (60,80]  (0,20]   (40,60]  (60,80]  (40,60]
 [31] (20,40]  (80,100] (80,100] (80,100] (80,100]
 [36] (40,60]  (60,80]  (60,80]  (40,60]  (40,60]
 [41] (0,20]   (60,80]  (40,60]  (40,60]  (80,100]
 [46] (80,100] (20,40]  (20,40]  (80,100] (40,60]
```

위의 명령어를 한 번 해 보세요. cut이라는 명령어는 구간으로 나누어 주는 명령어 입니다. example에 있는 자료를 break에서 정의한 구간에 따라 나누어 주는 것이지요. 그 결과를 interval 이라는 변수에 저장합니다. interval을 쳐서 그 안에 무엇이 들어있는지 볼까요? example에 저장되어 있었던 각각의 숫자들이 어느 구간에 포함되는지를 보여주고 있습니다. example에는 총 200개의 데이터가 있었으니, 이 구간표시 값도 200개가 되겠네요. 이것을 이용해서 각 구간의 빈도를 세면 되겠네요. 빈도를 세는 함수는 table입니다.

```
>
> table(interval)
interval
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
      36       48       38       40       38
>
```

어때요? 어렵지 않죠?

프로그램을 잘 하는 사람들은 이 모든 것을 한 줄의 명령어로 입력하기도 합니다.

```
>
> table(cut(sample(1:100, replace=T), breaks=c(0,20,40,60,80,100)))

  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
      18       15       20       19       28
>
```

위와 같이 모든 명령어를 한 줄에 담아 입력할 수도 있다는 말이지요. 괄호가 아주 중요합니다. 명령어의 영향이 어디까지 미치는지를 잘 정의해 주어야 하는 것이지요. 한 번 해 보세요.

자, 이렇게 도수분포표를 구해 보았습니다. 내용이 많아 보이지만, 별로 어렵지 않죠? 그러면 상대 도수나 누적도수는 어떻게 구할까요? 이 역시 R에서 제공하는 함수가 있습니다.

다음을 한 번 해 봅시다.

```
>
> freq <- table(interval)
>
> freq
interval
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
      36       48       38       40       38
>
> prop.table(freq)
interval
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
    0.18     0.24     0.19     0.20     0.19
>
> cumsum(freq)
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
      36       84      122      162      200
>
> cumsum(prop.table(freq))
  (0,20]  (20,40]  (40,60]  (60,80] (80,100]
    0.18     0.42     0.61     0.81     1.00
>
```

우선 table(interval)의 결과를 freq변수에 넣어 둡니다.

잘 들어 갔는지 한 번 살펴 보구요.

freq 변수에 prop.table이라는 함수를 사용해 주면, 상대도수가 나오지요. 전체에 대한 비율이 얼마인가에 해당하는 값입니다.

누적도수는 cumsum이라는 함수를 사용하여 구할 수 있어요.

누적상대도수는 어떻게 구할까요? 상대도수를 누적시키면 되니까, 두 개의 함수를 함께 사용해 주면 되겠지요.

자 이제, 몇가지 그림을 그려 볼까요?

히스토그램이나 줄기잎 그림은 원래 데이터를 그대로 사용해서 그리면 됩니다. 위에서 계속 했던 예제를 보면 example 변수에 들어 있는 데이터를 그대로 사용하면 된다는 것이지요. 위의 과정은 도수분포표를 만들어 보려고 다소 복잡한 단계를 거친 것인데, histogram이나 stem and leaf plot을 그릴 때는 명령어가 아주 간단합니다.

여러분이 이미 그려 본 파이차트는 어떨까요? 전체에서 해당 조각의 비율로 표시되는 그래프죠? 그러니 비율을 계산할 수 있는 데이터여야 합니다. example은 비율을 계산할 수 있는 데이터가 아니었어요. 구간도 나뉘어져 있지 않은 원자료(raw data)지요. 그러니, 파이챠트를 그리기 위해서

는 구간이 정의되어 있어야 하고, 각 구간의 비율을 계산할 수 있는 데이터가 있어야 합니다. 바로 freq 변수가 그런 데이터예요.

```
>
> hist(example)
>
> stem(example)

  The decimal point is 1 digit(s) to the right of the |

   0 | 22334444
   0 | 556678
   1 | 000023334
   1 | 5555677899
   2 | 00011234
   2 | 55567777778899999
   3 | 01111112344
   3 | 5555677778999
   4 | 001122244
   4 | 56677788899
   5 | 022224
   5 | 55666778888
   6 | 000111233344
   6 | 5566677778889999
   7 | 22334
   7 | 5667777889
   8 | 11224
   8 | 55567788889
   9 | 01122344
   9 | 5555777778889
  10 | 0

> pie(freq)
>
```

자, 위와 같이 한 번 해 봅시다. 그림 창에 예쁜 그림이 뜨지요?

이거 은근히 재미있답니다. ^^

이번에도 인증샷을 올려 주세요.

여러분들이 직접 해 보았다는 인증샷입니다.

여러분들이 새롭게 찾아서 추가한 기능이 있다면 본문에 간단히 적어서 소개해 주세요.

그리고, 설치과정을 포함해서, 잘 안되거나 에러가 발생하는 경우, 조교에게 쪽지로 보내지 말고, 캡쳐해서 게시판에 올려 주세요. 집단 지성을 활용합시다. 같은 문제를 가지고 해결책을 찾은 친구들이 답을 달아 주세요. 참고했던 웹사이트가 있으면 링크도 달아 주고... 에러가 생겼는데, 잘 해결된 경우에도 게시판에 올려주면 참고할 수 있는 사람들이 있을거예요. 여러분들의 적극적인 활동을 바랍니다.

나는 또 여러분의 창의적인 인증샷 기대합니다. 생각보다 재밌네요. 조금씩 보여주는 어려분의 모습이 참 멋집니다. 파이팅 !!