과제 보고서

(R-Py Computing Homework Final)

과목명	AI+X R-PY 컴퓨팅(AIX0004)
담당 교수님	이정환 교수님
제출일	2019년 12월 23일(월요일)
소속	한양대학교 공과대학 컴퓨터소프트웨어학부
학번	**이름**
2019009261	최가온(CHOI GA ON)

-목차-

Ⅰ. Part 1: 시뮬레이션을 통한 옵션 가격 설정

콜옵션은 기준자산을 만기일에 행사가격을 주고 살 수 있는 권리이다. 이때 만기일의 주식시장의 상태에 따라, 콜옵션 매입자는 만기 기준자산의 가격과 행사가격 차이만큼의 수입을 얻는다. 이때 기준자산의 가격이 행사가격보다 작으면, 아무런 수입도 얻지 못하게 되는 것이다. 요약하면, 매입자는 만기의 기준자산 가격이 행사가격을 초과할 경우에만 그 옵션을 행사(Exercise)하게 된다.

1. **문제 1** - 수입 Y를 S_T의 가격 그래프로 나타내기

만기에서의 콜옵션 소유자의 수입 Y를 기준자산, 여기서는 주식 S_T의 가격의 그래프로 나타내라. S_T의 범위는 0부터 200까지이며 K=100으로 고정한다.

콜옵션 소유자의 입장에서 만기의 수입 Y는 다음과 같다.

$$Y = \max(S_T - K, 0)$$

이를 바탕으로 R을 이용하여 프로그래밍을 하였다.

먼저, S_T를 price 변수로 표현하였는데, 그것의 범위가 0부터 200까지이므로 이는 정수형 벡터로 표현하였다. 또한 S_T의 각각의 원소에 대해 Y의 값이 하나씩 대응되게 되는 구조이므로, Y는 201개의 원소(0부터 200까지는 201개이다.)가 할당된 벡터로 구현하였다. for 구문에서는 Y 각각의 원소에 위에 제시된 수식을 적용하여 리턴된 값을 저장하게 된다.

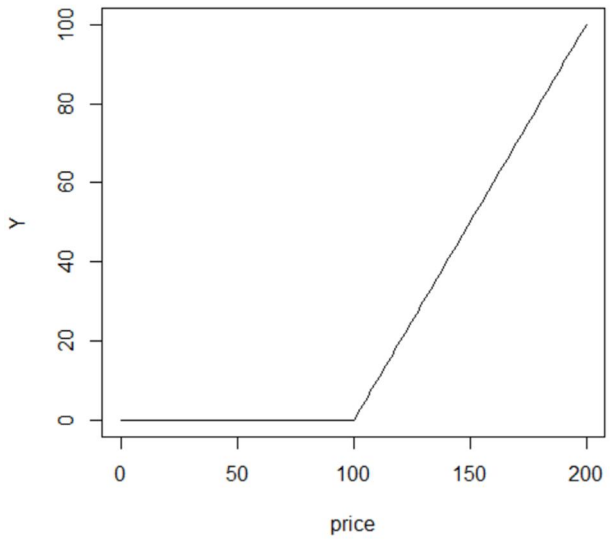
마지막으로, x축을 price, y축에 Y를 갖는 Cartesian 평면 위에 표현하여 그것의 Plot을 출력하였다.

2. 문제 1의 전체 코드

```
price=c(0:200)
Y=vector("numeric", 201)
for(i in 1:201) {
```

```
    Y[i]=max(price[i]-100, 0)
}
plot(price,Y,type='l')
```

(실행결과)



3. **문제 2** - 블랙-숄츠-머턴 방정식 구현

블랙, 숄츠, 머턴은 콜옵션의 프리미엄 C가 특정가정들을 만족하면 다음과 같은 식에 의해 결정된다고 하였다.

$$C = S \cdot N(d_1) - Ke^{-r_f \cdot T} \cdot N(d_2)$$

콜옵션의 프리미엄 C가 특정가정들을 만족하면 다음과 같은 식에 의해 결정된다고 하였다.

4. 문제 2의 전체 코드

```
d_1<-function(S, K, r_f, σ, T) {
    (log(exp(S/K))+(r_f+(1/2)*σ*σ)*T)/(σ*sqrt(T))
    # S: 주식의 가격
```

```
    # K: 행사가격
    # r_f: 무위험이자율
    # σ: 주식수익률의 연간 표준편차
    # T: 옵션의 만기
}

d_2<-function(S, K, r_f, σ, T) {
    d_1(S, K, r_f, σ, T) - σ*sqrt(T)
}

N<-function(num) {
    pnorm(num, mean = 0, sd = 1)
}

C<-function(S, K, r_f, σ, T) {
    S * N(d_1(S, K, r_f, σ, T)) - K * exp((-1) * r_f * T) * N(d_2(S, K, r_f, σ, T))
}

C(100, 100, 0.04, 0.1, 0.5)
```

(실행결과)

```
[1] 1.980133
```

5. 문제 3 - 주가의 경로 시뮬레이션 및 옵션의 가치 그래프

먼저 그래프를 그려야하므로 아래의 코드를 통해 "ggplot2" 패키지를 불러온다.

```
library("ggplot2")
```

그리고, 문제에서 제시된 상황에 대한 변수를 선언하고 초기화한다.

```
S0 <- 100
r_f <- 0.04
T <- 0.5
dt <- 0.001
N <- 10000
sigma <- 0.1
```

아래의 S는 주식의 가격을 의미한다. N번 시뮬레이션을 하기 위해 N개의 행을 가지고 있고, 아주 작은 시간(dt) 단위로 주식의 가격을 계산하기 위해 (T/dt) + 1개의 열을 만들었다. 이때 주식 가격의 초기값이 S0이므로 이를 포함시키기 위해 열의 크기는 "+1"을 하였다.

```
S <- matrix(0L, nrow = N, ncol = (T/dt) + 1)
S[, 1] <- matrix(S0, nrow = N, ncol = 1)
for (x in 1:N) {
for (y in 1:(T/dt)) {
S[x, y+1] <- S[x, y] * exp((r_f - 0.5 * sigma*sigma) * dt + sigma * sqrt(dt) * rnorm(1, 0,
1))
}
}
```

X는 시간 단위를 나타내기 위해 만들었다.

```
S <- matrix(0L, nrow = N, ncol = (T/dt) + 1)
S[, 1] <- matrix(S0, nrow = N, ncol = 1)
for (x in 1:N) {
for (y in 1:(T/dt)) {
S[x, y+1] <- S[x, y] * exp((r_f - 0.5 * sigma*sigma) * dt + sigma * sqrt(dt) * rnorm(1, 0,
1))
}
}
```

만기 시점에서의 옵션의 가치를 벡터의 형태로 저장하기 위해 V를 만들었다. 이때 옵션의 가치는 아래의 수식을 이용하여 결정하였다.

$$V^i = \max(S_T^i - K, 0)$$

```
V <- matrix(0L, nrow = 1, ncol = N)
for (x in 1:N){
V[1, x] <- max(S[x, (T/dt) + 1] - K, 0)
}
```

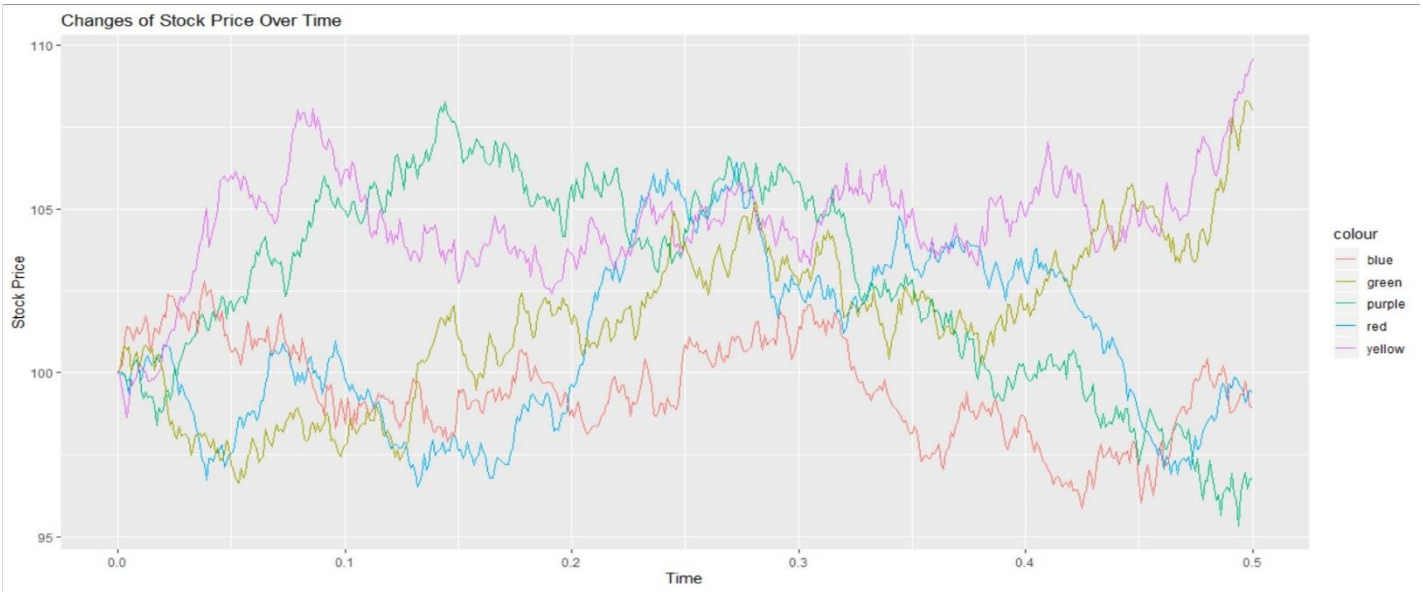
마지막으로, 옵션의 가치 C를 정하는 단계이다. 이는 아래의 수식을 이용하여 산출하였다.

$$C = \exp(-r_f T)\frac{1}{N}\sum_{i=1}^{N} V^i$$

아래는 그래프에 대한 코드이다. S에서 1번째부터 5번째까지의 Case만을 이용하였으며, 각각의 색깔을 달리하여 표시하였다. x축은 시간, y축은 주식의 가격을 의미하며, Plot의 제목은 "Changes of Stock Price Over Time"으로 하였다.

```
ggplot()+
geom_line(aes(x=X[1, ], y=S[1, ], color="red"))+
geom_line(aes(x=X[1,], y=S[2, ], color="purple"))+
geom_line(aes(x=X[1,], y=S[3, ], color="blue"))+
geom_line(aes(x=X[1,], y=S[4, ], color="yellow"))+
geom_line(aes(x=X[1,], y=S[5, ], color="green"))+
xlab("Time")+
ylab("Stock Price")+
ggtitle("Changes of Stock Price Over Time")
```

그래프는 아래와 같이 그려지게 된다.



6. 문제 3의 전체 코드

```
library("ggplot2")

S0 <- 100                    # 처음 주식의 가격
r_f <- 0.04                  # 무위험이자율
T <- 0.5                     # 옵션의 만기
dt <- 0.001                  # 아주 작은 시간(미분개념)
N <- 10000                   # 시뮬레이션 횟수
sigma <- 0.1                 # 주식수익률의 연간 표준편차
K <- 100                     # 행사 가격

S <- matrix(0L, nrow = N, ncol = (T/dt) + 1)
S[, 1] <- matrix(S0, nrow = N, ncol = 1)
for (x in 1:N) {
for (y in 1:(T/dt)) {
S[x, y+1] <- S[x, y] * exp((r_f - 0.5 * sigma*sigma) * dt + sigma * sqrt(dt) * rnorm(1, 0,
1))
}
}
```

```
X <- matrix(0L, nrow = 1, ncol = (T/dt) + 1)
for (x in 1:(T/dt)) {
X[1, x + 1] <- X[1, x] + dt
}

# 만기 시점의 옵션의 가치
V <- matrix(0L, nrow = 1, ncol = N)
for (x in 1:N){
V[1, x] <- max(S[x, (T/dt) + 1] - K, 0)
}

# 최종 옵션의 가치
C <- exp(-r_f * T) * sum(V) / N
print("옵션의 가치: ")
C

# 그래프 그리기
ggplot()+
geom_line(aes(x=X[1, ], y=S[1, ], color="red"))+
geom_line(aes(x=X[1, ], y=S[2, ], color="purple"))+
geom_line(aes(x=X[1, ], y=S[3, ], color="blue"))+
geom_line(aes(x=X[1, ], y=S[4, ], color="yellow"))+
geom_line(aes(x=X[1, ], y=S[5, ], color="green"))+
xlab("Time")+
ylab("Stock Price")+
ggtitle("Changes of Stock Price Over Time")
```

Ⅱ. Part 2: 개별 기말과제

1. 연구 문제와 연구의 목적

The Open University(오픈 대학교)는 영국의 학부 교육을 위한 공교육 대학이다. 이 대학교의 가장 큰 특징 중 하나는 거의 대부분의 교육이 캠퍼스라는 물리적인 공간에서 행해지는 것이 아니라는 점에 있다. 대학이 설립될 당시인 1969년에도 공공 텔레비전이나 편집 시설 등을 이용하여 교육물들을 만들어 나갔다. 현재 17만명이 넘는 학생들이 이 학교에 등록되어 있는 상태이며, 그 중 31%는 25세 이하이고 7400개가 넘는 지역의 해외 출신들로 대거 포집해있다. 이로 인해 세계에서 가장 큰 대학이라는 평가를 받고 있다는 점이 주목할 만하다.

한편, 현대 사회는 여전히 교육의 불평등의 문제가 만연하다. 한국 사회에서는 대학의 서열화는 물론이고, 미국이나 일본과 유사하게 사립학교의 비중이 타국에 비해 월등히 높은 편이다. 이에 따라 평균적인 등록금의 액수도 높은 편이며 이에 부담을 갖는 가계(학생)도 상당하다. 위의 오픈 대학교와 같은 온라인 형식의 학교가 이러한 사회 문제를 해결하기 위한 하나의 대안이 될 수 있다.

온라인 형식의 학교의 특징 중 가장 대표적인 것은 비용 절감이다. 보통 어떤 재화든 그 생산규모가 커지면 총평균비용이 감소하는 규모의 경제를 띠다가, 어느 수준 이상에 도달하면 반대로 총평균비용이 증가하는 규모의 불경제를 띠게 된다. 이는 현재의 생산 수준과 기술 그리고 생산요소의 사용 등이 좀 더 개선되어야 할 필요성을 시사하기도 한다. 반면, 인터넷 사회에서 발생하는 재화(게임물, 애니메이션, 그리고 지금 워드로 작성하고 있는 보고서까지)는 거의 대부분 규모의 경제의 모습만을 보이게 된다. 그 이유는 간단히 말해 재화의 생산량을 늘림에 있어서 비용이 0에 가깝다. 그저 재화를 인터넷의 서버에 올려놓기만 하면, 필요한 사람이 그것에 대한 비용을 지불하기만 하면 되기 때문이다.

위와 같은 이유로 오픈 대학교와 같은 사이버 기반의 학교 교육 시스템은 교육 불평등이라는 사회적인 문제뿐만 아니라 경제학적으로도 사회적 효용을 증가시키는 시스템이라고 할 수 있다.

이번 연구에서는, 이러한 사이버 기반의 학교교육 시스템에서 학습한 학생의 학업 성취도에 대한 분석이 주를 이룬다. 사이버 기반 시스템으로 인한 접근의 용이성으로 연령대, 문화권, 성별, 장애의 유무 등에서 학생들은 다양성을 지닌다. 한편으로는 학생의 스펙트럼이 넓게 나타나다보니, 그러한 학생들이 모두 만족스러운, 다시 말해 자신의 학업 성취에 긍정적인 영향을 줄 수 있는지에 대해서 연구해보아야 할 필요가 있다. 이번 연구에서는 학생의 특징(위에서 말한 특징)에 따른 학업 성취의 결과를 분석하여 사이버 기반의 학습 시스템의 효율을 향상시키는 방법에 대해 모색할 것이다.

2. 연구에서 사용한 데이터(JSON 형식)에 대한 기술

이번 연구에서는 학생들의 정보와 학업 성취의 결과가 기록된 JSON 파일을 이용할 것이다. 자료의 출처는 아래와 같다.

https://www.kaggle.com/rocki37/open-university-learning-analytics-dataset/version/1#vle.csv

위의 사이트에서 studentInfo.csv와 studentAssessment.csv 파일을 이용할 것이다. JSON 형식으로 데이터를 다루기 위해 분석 모형을 파이썬으로 구현하는 과정에서는 CSV 파일을 JSON 파일로 변환하여 연구에 사용하였다.

studentInfo.csv는 학생의 국적, 나이, 교육기간 등을 포함한 학생의 개인정보가 담긴 파일이고, studentAssessment.csv에는 학생이 Open University의 과정을 수료한 후의 점수 데이터를 포함하고 있다.

이때 StudentInfo.csv에 있는 학생정보와 studentAssessment.csv에 있는 학생점수를 학생의 고유 id를 기반으로 하여 서로 match시켰다. 이때 새롭게 생성된 파일명은 result.json이다. 통합된 이 파일을 이용하여 통계 분석을 진행한다. 그것에 대한 코드는 다음과 같다.

```
path1="C:\\Users\\choig\\Desktop\\파일\\1 학년 2 학기\\AIX R-PY 컴퓨팅\\과제\\최종 보고서\\Open University Learning Analytics Dataset\\studentInfo.json"
path2="C:\\Users\\choig\\Desktop\\파일\\1 학년 2 학기\\AIX R-PY 컴퓨팅\\과제\\최종 보고서\\Open University Learning Analytics Dataset\\studentAssessment.json"

import json

if __name__ == '__main__':
    assessment = dict()
    info = dict()
    lst = set()
```

```
with open(path2, 'r') as f:
    as_data = json.load(f)

    for asd in as_data:
        assessment[asd['id_student']] = asd
        lst.add(asd['id_student'])

with open(path1, 'r') as f:
    info_data = json.load(f)

    for infod in info_data:
        if infod['id_student'] not in lst:
            continue

        info[infod['id_student']] = infod

print(f'Total data count: {len(lst)}')

for id in lst:
    assessment[id].update(info[id])

with open('result.json', 'w') as f:
    json.dump([*assessment.values()], f)
```

각각의 데이터 feature에 대한 설명은 다음과 같다.(이번 실험에서 사용할 feature에 대해서만 서술함.)

(1) id_student

학생 개개인을 식별하기 위한 고유번호이다.

(2) code_module

수업의 학수번호에 해당한다.

(3) gender

학생의 성별을 의미하고, M(Male)과 F(Female)로 표시하였다.

(4) region

학생의 출신 지역을 의미한다.

(5) age_band

학생의 나이를 일정 범위로 쪼개어 표현한 것이다. "0-35", "35-55", "55<="의 세 집합으로 구성된다.

(6) disability

학생의 장애 유무를 표현한 것으로, "Y"와 "N"의 두 가지 값 중 하나를 갖는다.

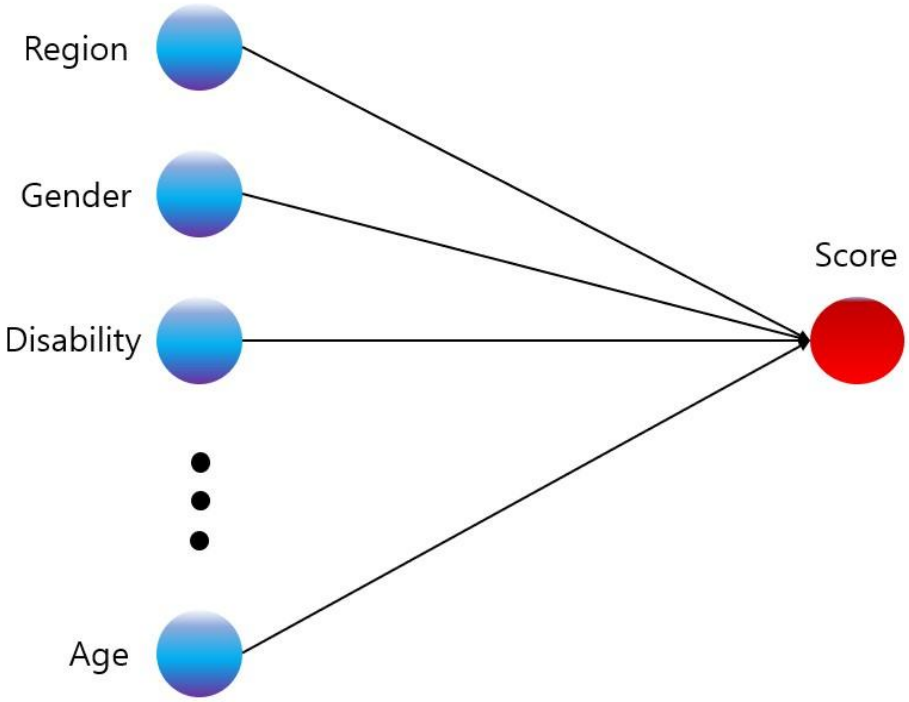
(7) final_result

간단히 말해, PF 형식이다. "Pass", "Withdrawn", "Fail"의 3가지 경우의 수를 가지는 feature이다.

(8) score

학생이 수업의 평가에서 얻은 최종 점수를 의미하며, 통상적으로 쓰이는 0부터 100까지의 자연수의 범위를 가진다.

3. 분석 모형 설정과 그것의 해석



이번 연구에서는 학생이 가진 특징 - 지역, 성별, 장애의 유무, 나이 등 - 으로 학생의 점수와의 상관관계를 파악할 것이다. 이번 연구의 목적은 학생 개인별 맞춤형 교육 시스템을 구축하는데 도움이 될 해석을 제시하는 것이다.

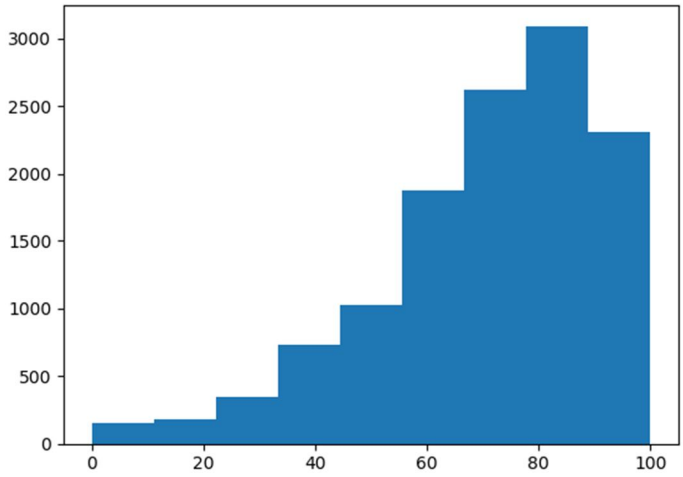
우선, 학생의 특징 1개씩 통계 분석을 한다. 즉, 학생의 지역에 따른 점수분포,

학생의 성별에 따른 점수분포, 학생의 장애유무에 따른 점수분포, 학생의 나이에 따른 점수분포를 확인하는 것이 이번 분석 모델의 특징이다.

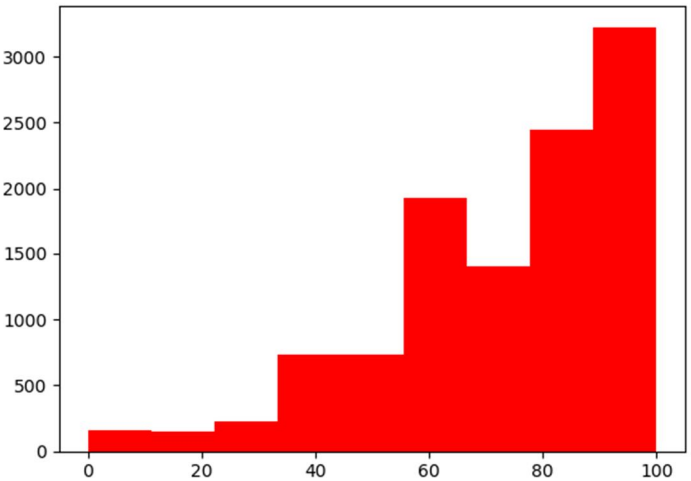
이때 각각의 상관관계는 히스토그램으로 표현할 것이다. JSON 파일을 가장 큰 데이터프레임의 형태로 만든 후, 그것의 일부를 Slicing하여 각각의 Feature별로 Sub-데이터프레임을 만든다. 이후 이 작게 쪼개진 데이터프레임을 기반으로 점수와 Feature의 상관관계를 시각적으로 표현할 히스토그램을 제시한다.

4. 모형 분석의 결과 및 해석

1) 성별에 따른 성적 분포 분석



[그림1] 남학생 성적분포



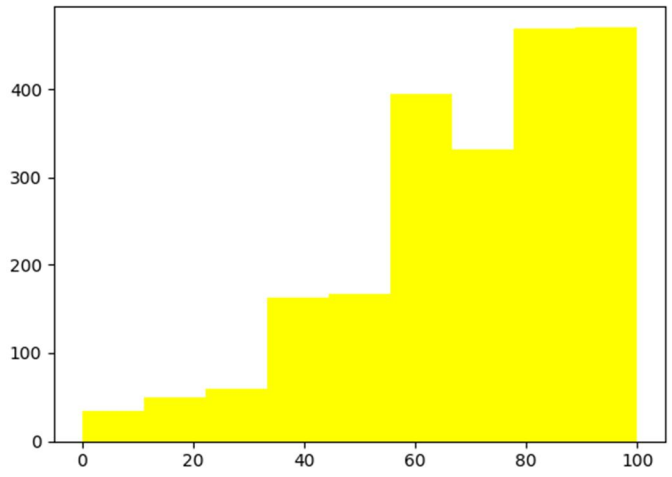
[그림2] 여학생 성적분포

남녀 모두 각각의 분포에서 0점~100점의 학생수가 모두 존재한다. 평균값은

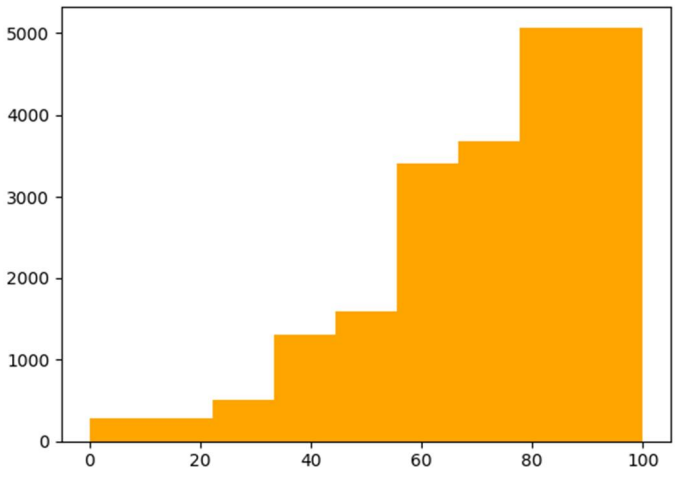
남학생이 여학생에 비해 다소 작은 편이다. 이때, 수업의 총점이 다소 상향평준화되어 대다수의 학생이 70점 이상의 점수를 획득하였다는 것을 알 수 있다.

2) 장애에 따른 성적 분포 분석

장애가 있는 학생만 따로 묶어 성적의 분포를 나타내는 히스토그램을 만들고, 이와 똑같은 방법으로 장애가 없는 일반학생을 대상으로 성적분포 그래프를 만들어볼 것이다. 우선, 학생수를 분석해보면, 약 2만 5천여 명의 학생중 장애인 학생은 2000명 정도에 해당하며, 10%가 약간 되지 않는 정도라는 것을 알 수 있다.



[그림3] 장애인 학생 성적분포



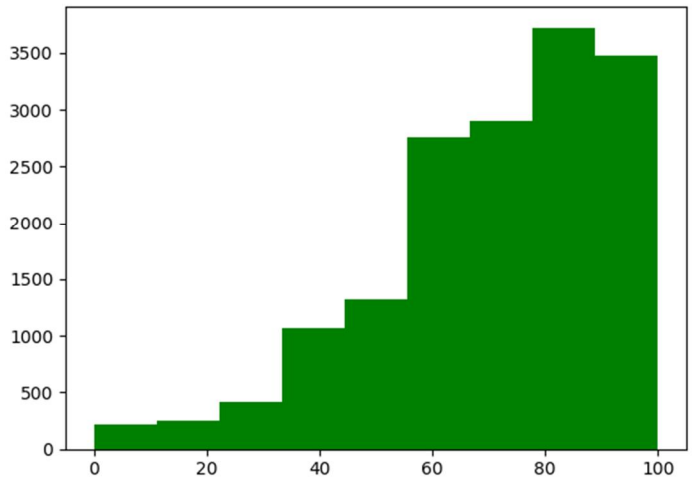
[그림4] 비장애인 학생 성적분포

성별로 나누었을 때와 마찬가지로, 80점~100점대의 학생수가 가장 많게 측정되었다. 약간 차이가 나는 부분은 60-70점에 해당하는 학생이 장애인 학생의 점수분포에서 다소 크게 나타났다는 점이다. 이를 통해 학생의 신체적 환경에 학생마다 차이가 있음에도 불구하고, Open University가 학생 개개인 맞춤별 교육 서비스를 제공하고 있다는 사실을 알 수 있다. 일반적으로, 오프라인 상에서 항상 그런 것은 아니지만 장애인 학생의 점수는 정규분포 상의 곡선에서 0.5, 0.6 상의 위치에서 나타나는 것이 대부분이나, 현재 그래프에서 보듯이 Open University에서는 장애인 학생과 비장애인 학생 간의 점수 차이가 크게 나지 않았음을 알 수 있다.

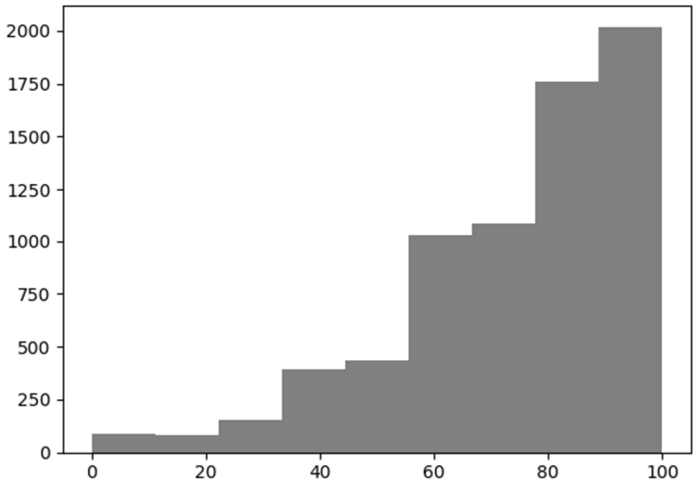
3) 나이에 따른 성적 분포 분석

20대의 학생들은 주로 대학교라는 교육 기관에서 교육을 받는 것이 일반적인 상황이다. 그리고 40대 이상의 사람들 중 교육을 원하는 사람에 한해 평생교육원, 대학원 등에서 교육 서비스를 제공하고 있다. 이때 Open University는 연령의 넓은 스펙트럼을 가지고 있어 위에서 말한 대학과 평생교육원의 역할을 동시에 하게될 것으로 기대할 수 있다. 따라서 이번 분석은 온라인 교육이 모든 연령대에서 실질적인 효과가 나타날지에 대해 알아보는 의미가 있다고 할 수 있다.

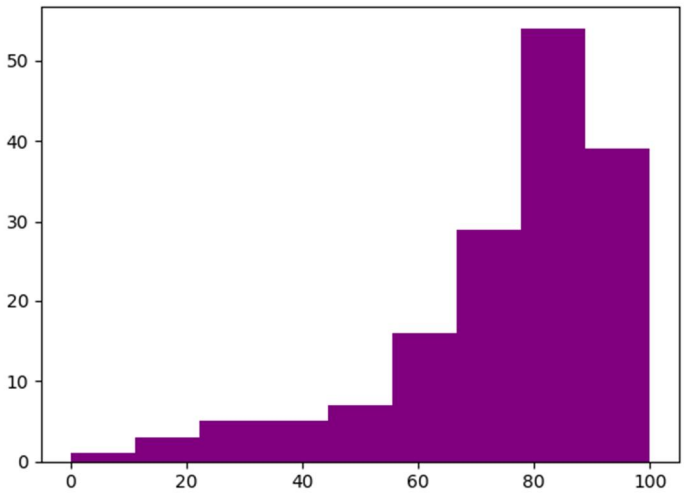
히스토그램을 분석한 결과, 나이대별에 상관없이 공통적으로 성적은 모두 70 ~ 100점대에 집중적으로 분포된 결과를 보였다.



[그림5] 0~35세 학생 성적분포



[그림6] 35~55세 학생 성적분포



[그림7] 55세 이상 학생 성적분포

학생의 나이가 증가하면 일반적으로 학생의 성적은 하락하게 될것이라 생각하는 것이 일반적이다. 그러나, 세 집단에서 높은 성적이 골고루 분포해있었다. 그러나, 55세 이상의 학생에서는 90 ~ 100점대의 학생이 현저히 낮아짐을 알 수 있었다.

4) 전체적인 분석 및 이번 연구의 한계점

어느 집단에서 관찰해보아도 0점~40점의 성적 구간의 학생이 다소 존재한다. 이 원인은 단순히 학생이 학업을 따라가는데의 어려움만이라고 보기는 어렵다고

생각한다. 우선, Open University는 일종의 공공재의 성격을 띠고 있다. 즉, 비용을 적게 지불해도 누구나 수강할 수 있는 시스템이기에 비싼 등록금을 지불하고 수강하는 대학교에 비해 수료율이 다소 낮아질 수 있다는 판단에서였다. 그럼에도, 학생 전체를 놓고 봤을 때에는 학업 성취도가 높은 편임을 확인할 수 있었다.

Open University에서 이뤄진 교육의 성과는 나름 학생의 개개인별 특징에 구애받지 않고 맞춤형 교육을 실현했다는 데에서 의미가 있다. 그러나, 연구에서 쓰이는 데이터에는 결측치가 다소 존재하였다. 이 결측치를 삭제하고, 일부 데이터를 가공하는 과정에서 모든 학생의 데이터를 볼 수 없었다는 점에서는 분명 한계가 있다. 또한 학생이 어떤 과목을 듣고 무슨 개념을 배웠는지 등에 대한 자료는 확실하게 존재하지가 않아 연구에 어려움이 있었다. 그럼에도 불구하고, 통계 분석을 통해 Open University의 실효성에 대해 증명해보였다는 점에서는 가치가 있다고 판단한다.

5. 전체 코드

```
import seaborn as sns
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

result_path = "C:\\Users\\choig\\Desktop\\파일\\1 학년 2 학기\\AIX R-PY
컴퓨팅\\과제\\최종 보고서\\result.json"
result = pd.read_json(result_path)

# 성별에 따른 성적분포
gender_score = result[['gender','score']]

male_score = pd.to_numeric(gender_score[gender_score['gender']=="M"]['score'])
female_score = gender_score[gender_score['gender']=="F"]

plt.hist(male_score.dropna(), np.linspace(0, 100, 10))
plt.show()

# 장애의 유무에 따른 성적분포
disability_score = result[['disability', 'score']]
```

```
disability_yes=pd.to_numeric(disability_score[disability_score['disability']=='Y']['score'])
disability_no=pd.to_numeric(disability_score[disability_score['disability']=='N']['score'])


plt.hist(disability_no.dropna(), np.linspace(0, 100, 10), color='orange')
plt.show()


# 나이에 따른 성적분포
age_score=result[['age_band', 'score']]

age1=pd.to_numeric(age_score[age_score['age_band']=="0-35"]['score'])
age2=pd.to_numeric(age_score[age_score['age_band']=="35-55"]['score'])
age3=pd.to_numeric(age_score[age_score['age_band']=="55<="]['score'])
plt.hist(age3.dropna(), np.linspace(0, 100, 10), color='purple')
plt.show()
```